In [1]:

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("factbook.db")
cursor = conn.cursor()
```

In [2]:

```
q = "select * from sqlite_master where type='table';"
cursor.execute(q).fetchall()
```

Out[2]:

```
[('table',
  'facts',
  'facts',
  2,
  'CREATE TABLE "facts" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "code" varchar(255) NOT
NULL, "name" varchar(255) NOT NULL, "area" integer, "area_land" integer, "area_water" integer, "po
pulation" integer, "population_growth" float, "birth_rate" float, "death_rate" float,
"migration_rate" float, "created_at" datetime, "updated_at" datetime)'),
 ('table',
  'sqlite_sequence',
  'sqlite_sequence',
  3,
  'CREATE TABLE sqlite_sequence(name,seq)')]
```

In [3]:

```
pd.read_sql_query(q, conn)
```

Out[3]:

	type	name	tbl_name	rootpage	sql
0	table	facts	facts	2	CREATE TABLE "facts" ("id" INTEGER PRIMARY KEY...
1	table	sqlite_sequence	sqlite_sequence	3	CREATE TABLE sqlite_sequence(name,seq)

In [4]:

```
q1 = "select * from facts limit 5"
cursor.execute(q1).fetchall()
```

Out[4]:

```
[(1,
  'af',
  'Afghanistan',
  652230,
  652230,
  0,
  32564342,
  2.32,
  38.57,
  13.89,
  1.51,
  '2015-11-01 13:19:49.461734',
  '2015-11-01 13:19:49.461734'),
 (2,
  'al',
  'Albania',
  28748,
  27398,
  1350,
  3029278,
  0.3,
  12.92,
  6.58,
  3.3,
  '2015-11-01 13:19:54.431082',
  '2015-11-01 13:19:54.431082'),
 (3,
  'ag',
  'Algeria',
  2381741,
  2381741,
  0,
  39542166,
  1.84,
  23.67,
  4.31,
  0.92,
  '2015-11-01 13:19:59.961286',
  '2015-11-01 13:19:59.961286'),
 (4,
  'an',
  'Andorra',
  468,
  468,
  0,
  85580,
  0.12,
  8.13,
  6.96,
  0.0,
  '2015-11-01 13:20:03.659945',
  '2015-11-01 13:20:03.659945'),
 (5,
  'ao',
  'Angola',
  1246700,
  1246700,
  0,
  19625353,
  2.78,
  38.78,
  11.49,
  0.46,
  '2015-11-01 13:20:08.625072',
  '2015-11-01 13:20:08.625072')]
```

In [5]:

```
pd.read_sql_query(q1, conn)
```

Out[5]:

	id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate	cr
0	1	af	Afghanistan	652230	652230	0	32564342	2.32	38.57	13.89	1.51	20 13:19:4
1	2	al	Albania	28748	27398	1350	3029278	0.30	12.92	6.58	3.30	20 13:19:5
2	3	ag	Algeria	2381741	2381741	0	39542166	1.84	23.67	4.31	0.92	20 13:19:5
3	4	an	Andorra	468	468	0	85580	0.12	8.13	6.96	0.00	20 13:20:0
4	5	ao	Angola	1246700	1246700	0	19625353	2.78	38.78	11.49	0.46	20 13:20:0

OUTLIERS

In [6]:

```
q2 = "select min(population), max(population), min(population_growth), max(population_growth) from
facts"
cursor.execute(q2).fetchall()
```

Out[6]:

```
[(0, 7256490011, 0.0, 4.02)]
```

In [7]:

```
pd.read_sql_query(q2, conn)
```

Out[7]:

	min(population)	max(population)	min(population_growth)	max(population_growth)
0	0	7256490011	0.0	4.02

Finding the country with population is 0

In [8]:

```
q3 = "select * from facts where population == 0"
cursor.execute(q3).fetchall()
```

Out[8]:

```
[(250,
  'ay',
  'Antarctica',
  None,
  280000,
  None,
  0,
  None,
  None,
  None,
  None,
  '2015-11-01 13:38:44.885746',
  '2015-11-01 13:38:44.885746')]
```

In [9]:

```
pd.read_sql_query(q3, conn)
```

Out[9]:

	id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate	creat
0	250	ay	Antarctica	None	280000	None	0	None	None	None	None	2015- 13:38:44.88

Since the min population is 0, select min population and set it equal to population instead set it directly to 0.

In [10]:

```
q4 = "select * from facts where population == (select min(population) from facts)"
cursor.execute(q4).fetchall()
```

Out[10]:

```
[(250,
  'ay',
  'Antarctica',
  None,
  280000,
  None,
  0,
  None,
  None,
  None,
  None,
  '2015-11-01 13:38:44.885746',
  '2015-11-01 13:38:44.885746',
  '2015-11-01 13:38:44.885746')]
```

In [11]:

```
pd.read_sql_query(q4, conn)
```

Out[11]:

	id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate	creat
0	250	ay	Antarctica	None	280000	None	0	None	None	None	None	2015- 13:38:44.8

In [12]:

```
q5 = "select * from facts where population == 7256490011"
cursor.execute(q5).fetchall()
```

Out[12]:

```
[(261,
  'xx',
  'World',
  None,
  None,
  None,
  7256490011,
  1.08,
  18.6,
  7.8,
  None,
  '2015-11-01 13:39:09.910721',
  '2015-11-01 13:39:09.910721')]
```

In [13]:

```
pd.read_sql_query(q5, conn)
```

Out[13]:

	id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate	created_
0	261	xx	World	None	None	None	7256490011	1.08	18.6	7.8	None	2015-11- 13:39:09.9107

In [14]:

```
q6 = "select * from facts where population == (select MAX(population) from facts)"
cursor.execute(q6).fetchall()
```

Out[14]:

```
[(261,
  'xx',
  'World',
  None,
  None,
  None,
  7256490011,
  1.08,
  18.6,
  7.8,
  None,
  '2015-11-01 13:39:09.910721',
  '2015-11-01 13:39:09.910721')]
```

In [15]:

```
pd.read_sql_query(q6, conn)
```

Out[15]:

	id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate	created_
0	261	xx	World	None	None	None	7256490011	1.08	18.6	7.8	None	2015-11- 13:39:09.9107

HISTOGRAM

In [16]:

```python
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)

q7 = "select population from facts where population != (select max(population) from facts) and pop
ulation != (select min(population) from facts)"
q8 = "select population_growth from facts"
q9 = "select birth_rate from facts"
q10 = "select death_rate from facts"

pd.read_sql_query(q7, conn).hist(ax=ax1)
pd.read_sql_query(q8, conn).hist(ax=ax2)
pd.read_sql_query(q9, conn).hist(ax=ax3)
pd.read_sql_query(q10, conn).hist(ax=ax4)
```

Out[16]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000022E96005B38>],
      dtype=object)
```

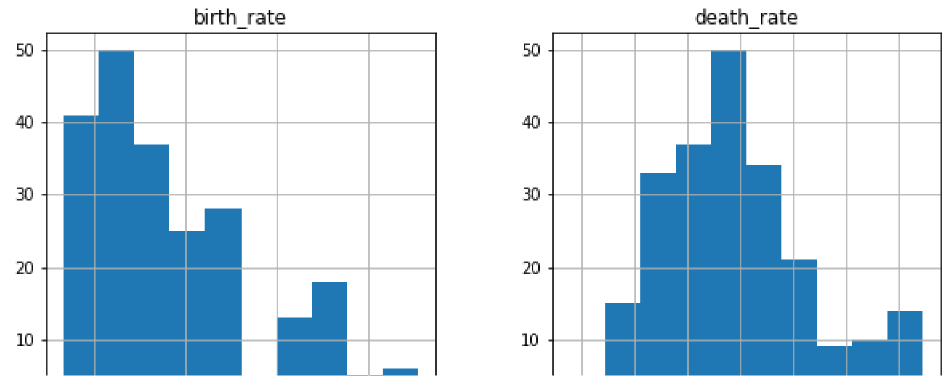
In [17]:

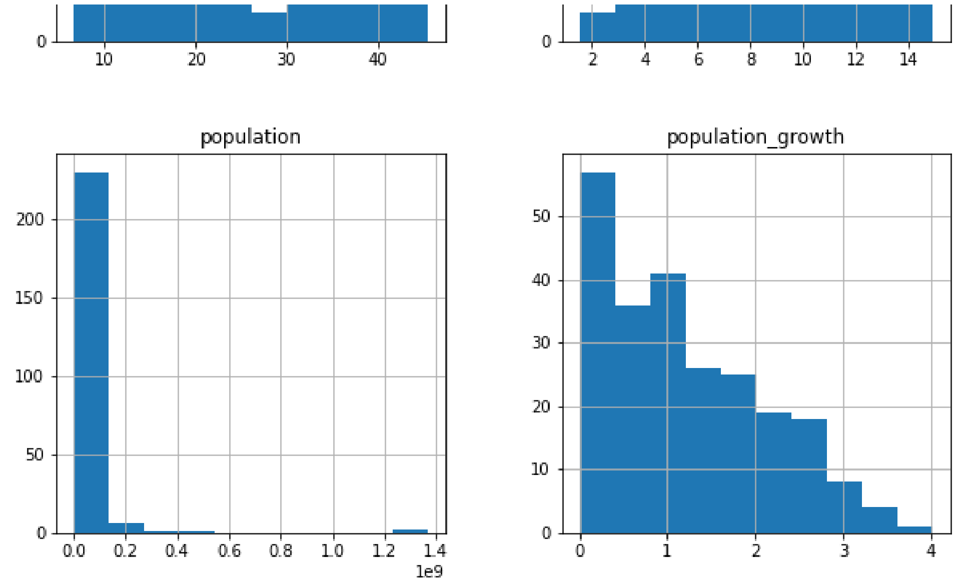
```python
fig1 = plt.figure(figsize = (10,10))
ax = fig1.add_subplot(111)
q11 = "select population, population_growth, birth_rate, death_rate from facts where population !=
(select max(population) from facts) and population != (select min(population) from facts)"
pd.read_sql_query(q11, conn).hist(ax=ax)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3325: UserWarning: To
output multiple subplots, the figure containing the passed axes is being cleared
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[17]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000022E9655D400>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022E9662D240>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000022E9665E7F0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022E96690DA0>]],
      dtype=object)
```





Which Countries have the highest population density?

In [18]:

```
q12 = "select name, cast(population as float)/cast(area as float) density from facts order by
density DESC"
pd.read_sql_query(q12, conn)
```

Out[18]:

	name	density
0	Macau	21168.964286
1	Monaco	15267.500000
2	Singapore	8141.279770
3	Hong Kong	6445.041516
4	Gaza Strip	5191.819444
5	Gibraltar	4876.333333
6	Bahrain	1771.859211
7	Maldives	1319.640940
8	Malta	1310.015823
9	Bermuda	1299.925926
10	Sint Maarten	1167.323529
11	Bangladesh	1138.069143
12	Guernsey	847.179487
13	Jersey	838.741379
14	Barbados	675.823256
15	Mauritius	656.777941
16	Taiwan	650.781712
17	Aruba	623.122222
18	Lebanon	594.682788
19	Saint Martin	588.037037
20	San Marino	541.311475
21	Korea, South	492.531047
22	Rwanda	480.740109
23	West Bank	475.318430
24	Nauru	454.285714
25	Tuvalu	418.038462

	name	density
25	Tuvalu	[illegible]
26	Netherlands	407.960523
27	Marshall Islands	398.845304
28	Israel	387.545209
29	Burundi	385.996263
...	...	...
231	Mongolia	1.913482
232	Pitcairn Islands	1.021277
233	Falkland Islands (Islas Malvinas)	0.276103
234	Svalbard	0.030172
235	Greenland	0.026653
236	Chad	NaN
237	Niger	NaN
238	Holy See (Vatican City)	NaN
239	Ashmore and Cartier Islands	NaN
240	Coral Sea Islands	NaN
241	Heard Island and McDonald Islands	NaN
242	Clipperton Island	NaN
243	French Southern and Antarctic Lands	NaN
244	Saint Barthelemy	NaN
245	Bouvet Island	NaN
246	Jan Mayen	NaN
247	British Indian Ocean Territory	NaN
248	South Georgia and South Sandwich Islands	NaN
249	Navassa Island	NaN
250	Wake Island	NaN
251	United States Pacific Island Wildlife Refuges	NaN
252	Antarctica	NaN
253	Paracel Islands	NaN
254	Spratly Islands	NaN
255	Arctic Ocean	NaN
256	Atlantic Ocean	NaN
257	Indian Ocean	NaN
258	Pacific Ocean	NaN
259	Southern Ocean	NaN
260	World	NaN

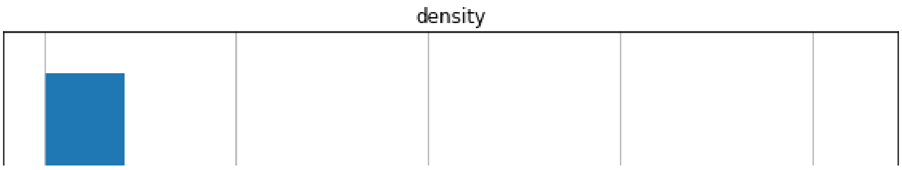
261 rows × 2 columns

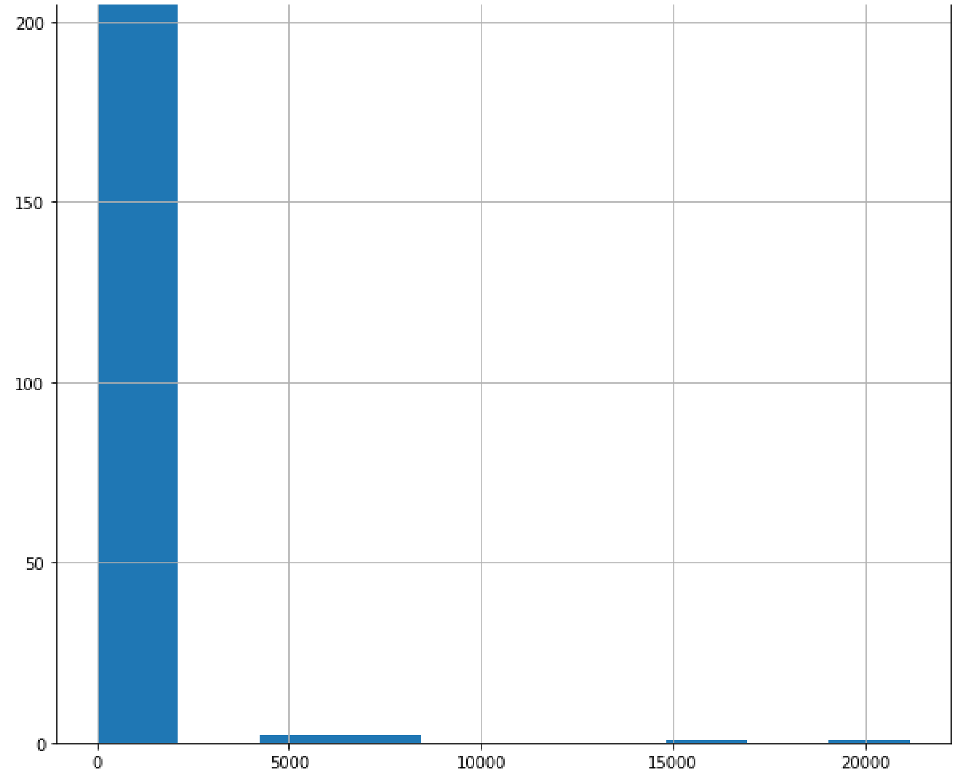
In [19]:

```
fig2 = plt.figure(figsize = (10,10))
ax = fig2.add_subplot(111)
pd.read_sql_query(q12, conn).hist(ax=ax)
```

Out[19]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000022E96A0A4A8>],
      dtype=object)
```





Which countries have the highest ratios of water to land?

In [31]:

```
q13 = "select name, cast(area_water as float)/cast(area_land as float) water_to_land from facts or
der by water_to_land DESC"
pd.read_sql_query(q13, conn)
```

Out[31]:

	name	water_to_land
0	British Indian Ocean Territory	905.666667
1	Virgin Islands	4.520231
2	Puerto Rico	0.554791
3	Bahamas, The	0.386613
4	Guinea-Bissau	0.284673
5	Malawi	0.259396
6	Netherlands	0.225710
7	Uganda	0.222922
8	Eritrea	0.164356
9	Liberia	0.156240
10	Bangladesh	0.140509
11	Gambia, The	0.116601
12	Taiwan	0.115313
13	Finland	0.112996
14	India	0.105634
15	Canada	0.098000
16	Sweden	0.097384
17	Colombia	0.096476
18	Brunei	0.094967
19	Guyana	0.092050
20	French Polynesia	0.088842

	name	water_to_land
21	Nicaragua	0.086507
22	Burundi	0.083723
23	Iran	0.076130
24	United States	0.072551
25	Tanzania	0.069429
26	Vietnam	0.068178
27	Rwanda	0.067699
28	Estonia	0.067000
29	Norway	0.064151
...	...	...
231	Turks and Caicos Islands	0.000000
232	American Samoa	0.000000
233	Guam	0.000000
234	Navassa Island	0.000000
235	Northern Mariana Islands	0.000000
236	Wake Island	0.000000
237	Gaza Strip	0.000000
238	Paracel Islands	0.000000
239	Spratly Islands	0.000000
240	Western Sahara	0.000000
241	Ethiopia	NaN
242	New Zealand	NaN
243	South Sudan	NaN
244	Sudan	NaN
245	Holy See (Vatican City)	NaN
246	European Union	NaN
247	Greenland	NaN
248	French Southern and Antarctic Lands	NaN
249	Saint Barthelemy	NaN
250	Saint Martin	NaN
251	Akrotiri	NaN
252	Dhekelia	NaN
253	United States Pacific Island Wildlife Refuges	NaN
254	Antarctica	NaN
255	Arctic Ocean	NaN
256	Atlantic Ocean	NaN
257	Indian Ocean	NaN
258	Pacific Ocean	NaN
259	Southern Ocean	NaN
260	World	NaN

261 rows × 2 columns

British Indian Ocean Territory has more water than land.