

In [170]:

```
1 data.mean()
```

Out[170]:

5.375

In [171]:

```
1 data.median()
```

Out[171]:

5.25

In [172]:

```
1 data.fillna(data.mean())
```

Out[172]:

```
0    1.000
1    5.375
2    3.500
3    5.375
4    7.000
5   10.000
dtype: float64
```

In [173]:

```
1 data.fillna(data.median())
```

Out[173]:

```
0    1.00
1    5.25
2    3.50
3    5.25
4    7.00
5   10.00
dtype: float64
```

In []:

```
1
```

Data Cleansing and preperation

In [1]:

```
1 import numpy as np
2 import pandas as pd
```

In [2]:

```
1 string_data = pd.Series(["aardvark", "alpha", np.nan, "avacado"])
```

In [3]:

```
1 print(string_data)
```

```
0    aardvark
1      alpha
2         NaN
3    avacado
dtype: object
```

In [4]:

```
1 string_data.isnull() #this is used to check which records are null and which records are not null in boolean
```

Out[4]:

```
0    False
1    False
2     True
3    False
dtype: bool
```

In []:

```
1
```

In [5]:

```
1 ser1 = pd.Series([1, np.nan, 3.5, np.nan])
```

In [6]:

```
1 ser1
```

Out[6]:

```
0    1.0
1    NaN
2    3.5
3    NaN
dtype: float64
```

In [7]:

```
1 ser1.dropna()
```

Out[7]:

```
0    1.0
2    3.5
dtype: float64
```

In []:

```
1
```

In [9]:

```
1 data = pd.DataFrame([[1,6.5,3],[1,np.nan,np.nan],[1,np.nan,np.nan],[np.nan,np.nan,np.nan],[np.nan,6,5,3]])
```

In [16]:

```
1 data
```

Out[16]:

	0	1	2	3
0	1.0	6.5	3.0	NaN
1	1.0	NaN	NaN	NaN
2	1.0	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN
4	NaN	6.0	5.0	3.0

In [17]:

```
1 data.dropna()
```

Out[17]:

	0	1	2	3
0	1.0	6.5	3.0	NaN

In [14]:

```
1 new = data.drop([3],axis=1)
```

In [15]:

```
1 new
```

Out[15]:

	0	1	2
0	1.0	6.5	3.0
1	1.0	NaN	NaN
2	1.0	NaN	NaN
3	NaN	NaN	NaN
4	NaN	6.0	5.0

In [18]:

```
1 new.dropna()
```

Out[18]:

	0	1	2
0	1.0	6.5	3.0

In []:

```
1
```

In [19]:

1 new

Out[19]:

	0	1	2
0	1.0	6.5	3.0
1	1.0	NaN	NaN
2	1.0	NaN	NaN
3	NaN	NaN	NaN
4	NaN	6.0	5.0

In [20]:

1 new.dropna(how='all') #row wise

Out[20]:

	0	1	2
0	1.0	6.5	3.0
1	1.0	NaN	NaN
2	1.0	NaN	NaN
4	NaN	6.0	5.0

In [22]:

1 new.dropna(axis=1,how='all') #col wise

Out[22]:

	0	1	2
0	1.0	6.5	3.0
1	1.0	NaN	NaN
2	1.0	NaN	NaN
3	NaN	NaN	NaN
4	NaN	6.0	5.0

In []:

1

In [23]:

1 df = pd.DataFrame(np.random.randn(7,3))

In [27]:

1 df

Out[27]:

	0	1	2
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [28]:

1 df.iloc[0:3] = np.nan

In [29]:

```
1 df
```

Out[29]:

	0	1	2
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [30]:

```
1 df.fillna(0)
```

Out[30]:

	0	1	2
0	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [31]:

```
1 df
```

Out[31]:

	0	1	2
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [35]:

```
1 df
```

Out[35]:

	0	1	2
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [37]:

```
1 df.fillna(df.mean())
```

Out[37]:

	0	1	2
0	0.011745	0.732031	0.228362
1	0.011745	0.732031	0.228362
2	0.011745	0.732031	0.228362
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In [36]:

```
1 df.fillna(df.median())
```

Out[36]:

	0	1	2
0	-0.071661	0.723594	0.153791
1	-0.071661	0.723594	0.153791
2	-0.071661	0.723594	0.153791
3	-0.452132	1.668369	0.192867
4	-0.426402	0.152779	0.910550
5	0.283081	1.294409	0.114716
6	0.642434	-0.187433	-0.304684

In []:

```
1
```

removing duplicates

In [39]:

```
1 data = pd.DataFrame(  
2     {'k1':['one','two']*3 + ['two'],  
3     'k2':[1,1,2,3,3,4,4]  
4     })
```

In [40]:

```
1 data
```

Out[40]:

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4
6	two	4

In [41]:

```
1 data.duplicated()
```

Out[41]:

0 False
1 False
2 False
3 False
4 False
5 False
6 True
dtype: bool

In [42]:

```
1 data.drop_duplicates()
```

Out[42]:

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4

In [43]:

```
1 data
```

Out[43]:

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4
6	two	4

In [44]:

```
1 data.drop_duplicates(['k1'])
```

Out[44]:

	k1	k2
0	one	1
1	two	1

In [45]:

```
1 data['v1'] = range(7)
```

In [46]:

```
1 data
```

Out[46]:

	k1	k2	v1
0	one	1	0
1	two	1	1
2	one	2	2
3	two	3	3
4	one	3	4
5	two	4	5
6	two	4	6

In [47]:

```
1 data.drop_duplicates(['k1', 'k2'])
```

Out[47]:

	k1	k2	v1
0	one	1	0
1	two	1	1
2	one	2	2
3	two	3	3
4	one	3	4
5	two	4	5

In []:

```
1
```

Transforming Data using Function or Mapping

In [48]:

```
1 data = pd.DataFrame(  
2     {  
3         "food": [  
4             'apple', 'egg', 'wheat', 'bread', 'rice', 'dal', 'grapes', 'pineapple', 'honey'  
5         ],  
6         "calories": [4,3,12,6,7.5,8,3,5,6]  
7     }  
8 )  
9
```

In [49]:

```
1 data
```

Out[49]:

	food	calories
0	apple	4.0
1	egg	3.0
2	wheat	12.0
3	bread	6.0
4	rice	7.5
5	dal	8.0
6	grapes	3.0
7	pineapple	5.0
8	honey	6.0

In []:

```
1
```

In [50]:

```
1 food_categories = {  
2     "apple": 'fruit',  
3     "egg": 'non veg',  
4     "wheat": 'grain',  
5     "bread": 'veg food',  
6     "rice": 'grain',  
7     "dal": 'grain',  
8     "grapes": 'fruit',  
9     "pineapple": 'fruit',  
10    "honey": 'veg food'  
11 }
```

In [51]:

```
1 food_categories
```

Out[51]:

```
{'apple': 'fruit',  
'egg': 'non veg',  
'wheat': 'grain',  
'bread': 'veg food',  
'rice': 'grain',  
'dal': 'grain',  
'grapes': 'fruit',  
'pineapple': 'fruit',  
'honey': 'veg food'}
```

In [52]:

```
1 data
```

Out[52]:

	food	calories
0	apple	4.0
1	egg	3.0
2	wheat	12.0
3	bread	6.0
4	rice	7.5
5	dal	8.0
6	grapes	3.0
7	pineapple	5.0
8	honey	6.0

In [56]:

```
1 type(data['food'][0])
```

Out[56]:

str

In [68]:

```
1 data['food'] = data['food'].str.lower()
```

In [69]:

```
1 data
```

Out[69]:

	food	calories	FoodCategory
0	apple	4.0	NaN
1	egg	3.0	NaN
2	wheat	12.0	NaN
3	bread	6.0	NaN
4	rice	7.5	NaN
5	dal	8.0	NaN
6	grapes	3.0	NaN
7	pineapple	5.0	NaN
8	honey	6.0	NaN

In [70]:

```
1 data['FoodCategory'] = data['food'].map(food_categories)
```

In [71]:

```
1 data
```

Out[71]:

	food	calories	FoodCategory
0	apple	4.0	fruit
1	egg	3.0	non veg
2	wheat	12.0	grain
3	bread	6.0	veg food
4	rice	7.5	grain
5	dal	8.0	grain
6	grapes	3.0	fruit
7	pineapple	5.0	fruit
8	honey	6.0	veg food

In []:

```
1
```

binning

In [72]:

```
1 age = [20,22,25,27,21,23,37,31,61,45,41,32]
```

In [73]:

```
1 bins = [18,25,35,60,100]
```

In [74]:

```
1 category = pd.cut(age,bins)
```

In [75]:

```
1 category
```

Out[75]:

```
[(18, 25], (18, 25], (18, 25], (25, 35], (18, 25], ..., (25, 35], (60, 100], (35, 60], (35, 60], (25, 35]]  
Length: 12  
Categories (4, interval[int64]): [(18, 25] < (25, 35] < (35, 60] < (60, 100]]
```

In [76]:

```
1 pd.value_counts(category)
```

Out[76]:

```
(18, 25]      5  
(25, 35]      3  
(35, 60]      3  
(60, 100]     1  
dtype: int64
```

In []:

```
1
```

categorical data in pandas

In [77]:

```
1 fruits = ['apple','mango','apple','apple']*2
```

In [79]:

```
1 fruits
```

Out[79]:

```
['apple', 'mango', 'apple', 'apple', 'apple', 'mango', 'apple', 'apple']
```

In [80]:

```
1 n = len(fruits)  
2 n
```

Out[80]:

```
8
```

In [81]:

```
1 df = pd.DataFrame({  
2     'fruit': fruits,  
3     'basket_id': np.arange(n),  
4     'count': np.random.randint(3,15,size=n),  
5     'weight': np.random.uniform(0,4,size=n)  
6 },  
7     columns=['basket_id', 'fruit', 'count', 'weight']  
8 )
```

In [82]:

```
1 df
```

Out[82]:

	basket_id	fruit	count	weight
0	0	apple	5	2.480422
1	1	mango	9	2.673737
2	2	apple	6	0.122506
3	3	apple	6	1.299618
4	4	apple	6	0.750015
5	5	mango	6	3.079911
6	6	apple	8	3.511744
7	7	apple	9	0.819876

In [85]:

```
1 df['fruit']
```

Out[85]:

```
0    apple
1    mango
2    apple
3    apple
4    apple
5    mango
6    apple
7    apple
Name: fruit, dtype: object
```

In []:

```
1
```

In [87]:

```
1 fruit_cat = df['fruit'].astype('category')
```

In [88]:

```
1 fruit_cat
```

Out[88]:

```
0    apple
1    mango
2    apple
3    apple
4    apple
5    mango
6    apple
7    apple
Name: fruit, dtype: category
Categories (2, object): ['apple', 'mango']
```

In [91]:

```
1 res = fruit_cat.values
```

In [92]:

```
1 res
```

Out[92]:

```
['apple', 'mango', 'apple', 'apple', 'apple', 'mango', 'apple', 'apple']
Categories (2, object): ['apple', 'mango']
```

In [93]:

```
1 res.categories
```

Out[93]:

```
Index(['apple', 'mango'], dtype='object')
```

In [94]:

```
1 res.codes
```

Out[94]:

```
array([0, 1, 0, 0, 0, 1, 0, 0], dtype=int8)
```

In []:

1

In [95]:

1 res.value_counts()

Out[95]:

```
apple    6
mango    2
dtype: int64
```

In []:

1

group by

In [97]:

```
1 df = pd.DataFrame({
2     'key': ['a', 'b', 'c']*4,
3     'value': np.arange(0, 12)
4 })
```

In [98]:

1 df

Out[98]:

	key	value
0	a	0
1	b	1
2	c	2
3	a	3
4	b	4
5	c	5
6	a	6
7	b	7
8	c	8
9	a	9
10	b	10
11	c	11

In [99]:

1 res = df.groupby('key').value

In [100]:

1 res

Out[100]:

```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x7f733e1666a0>
```

In [101]:

1 res.max()

Out[101]:

```
key
a     9
b    10
c    11
Name: value, dtype: int64
```

In [102]:

1 res.count()

Out[102]:

```
key
a     4
b     4
c     4
Name: value, dtype: int64
```

In [103]:

```
1 res.min()
```

Out[103]:

key
a 0
b 1
c 2
Name: value, dtype: int64

In [104]:

```
1 res.mean()
```

Out[104]:

key
a 4.5
b 5.5
c 6.5
Name: value, dtype: float64

In [105]:

```
1 res.median()
```

Out[105]:

key
a 4.5
b 5.5
c 6.5
Name: value, dtype: float64

In [106]:

```
1 res.sum()
```

Out[106]:

key
a 18
b 22
c 26
Name: value, dtype: int64

In []:

```
1
```

In [107]:

```
1 url = 'https://raw.githubusercontent.com/justmarkham/pandas-videos/master/data/drinks.csv'
```

In [108]:

```
1 df = pd.read_csv(url)
```

In [109]:

```
1 df
```

Out[109]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe
4	Angola	217	57	45	5.9	Africa
...
188	Venezuela	333	100	3	7.7	South America
189	Vietnam	111	2	1	2.0	Asia
190	Yemen	6	0	0	0.1	Asia
191	Zambia	32	19	4	2.5	Africa
192	Zimbabwe	64	18	4	4.7	Africa

193 rows × 6 columns

In []:

```
1
```

In [110]:

```
1 df.head()
```

Out[110]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe
4	Angola	217	57	45	5.9	Africa

In [111]:

```
1 df.tail()
```

Out[111]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
188	Venezuela	333	100	3	7.7	South America
189	Vietnam	111	2	1	2.0	Asia
190	Yemen	6	0	0	0.1	Asia
191	Zambia	32	19	4	2.5	Africa
192	Zimbabwe	64	18	4	4.7	Africa

In []:

```
1
```

In [113]:

```
1 df['beer_servings'].mean()
```

Out[113]:

106.16062176165804

In [114]:

```
1 df['wine_servings'].mean()
```

Out[114]:

49.45077720207254

In []:

```
1
```

In [115]:

```
1 df['continent']
```

Out[115]:

```
0      Asia
1    Europe
2    Africa
3    Europe
4    Africa
...
188  South America
189      Asia
190      Asia
191    Africa
192    Africa
Name: continent, Length: 193, dtype: object
```

In []:

```
1
```

In [118]:

```
1 df.groupby('continent').beer_servings.mean()
```

Out[118]:

```
continent
Africa      61.471698
Asia        37.045455
Europe      193.777778
North America 145.434783
Oceania      89.687500
South America 175.083333
Name: beer_servings, dtype: float64
```

In [117]:

```
1 df.groupby('continent').wine_servings.mean()
```

Out[117]:

```
continent
Africa      16.264151
Asia         9.068182
Europe     142.222222
North America 24.521739
Oceania     35.625000
South America 62.416667
Name: wine_servings, dtype: float64
```

In []:

```
1
```

In [119]:

```
1 df
```

Out[119]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe
4	Angola	217	57	45	5.9	Africa
...
188	Venezuela	333	100	3	7.7	South America
189	Vietnam	111	2	1	2.0	Asia
190	Yemen	6	0	0	0.1	Asia
191	Zambia	32	19	4	2.5	Africa
192	Zimbabwe	64	18	4	4.7	Africa

193 rows × 6 columns

In [120]:

```
1 df['continent'] == 'Asia'
```

Out[120]:

```
0      True
1     False
2     False
3     False
4     False
...
188    False
189     True
190     True
191    False
192    False
Name: continent, Length: 193, dtype: bool
```

In [122]:

```
1 asia = df[df['continent'] == 'Asia']
```

In [123]:

1 asia

Out[123]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
12	Bahrain	42	63	7	2.0	Asia
13	Bangladesh	0	0	0	0.0	Asia
19	Bhutan	23	0	0	0.4	Asia
24	Brunei	31	2	1	0.6	Asia
30	Cambodia	57	65	1	2.2	Asia
36	China	79	192	8	5.0	Asia
46	North Korea	0	0	0	0.0	Asia
77	India	9	114	0	2.2	Asia
78	Indonesia	5	1	0	0.1	Asia
79	Iran	0	0	0	0.0	Asia
80	Iraq	9	3	0	0.2	Asia
82	Israel	63	69	9	2.5	Asia
85	Japan	77	202	16	7.0	Asia
86	Jordan	6	21	1	0.5	Asia
87	Kazakhstan	124	246	12	6.8	Asia
90	Kuwait	0	0	0	0.0	Asia
91	Kyrgyzstan	31	97	6	2.4	Asia
92	Laos	62	0	123	6.2	Asia
94	Lebanon	20	55	31	1.9	Asia
102	Malaysia	13	4	0	0.3	Asia
103	Maldives	0	0	0	0.0	Asia
112	Mongolia	77	189	8	4.9	Asia
116	Myanmar	5	1	0	0.1	Asia
119	Nepal	5	6	0	0.2	Asia
127	Oman	22	16	1	0.7	Asia
128	Pakistan	0	0	0	0.0	Asia
134	Philippines	71	186	1	4.6	Asia
137	Qatar	1	42	7	0.9	Asia
138	South Korea	140	16	9	9.8	Asia
141	Russian Federation	247	326	73	11.5	Asia
149	Saudi Arabia	0	5	0	0.1	Asia
154	Singapore	60	12	11	1.5	Asia
161	Sri Lanka	16	104	0	2.2	Asia
167	Syria	5	35	16	1.0	Asia
168	Tajikistan	2	15	0	0.3	Asia
169	Thailand	99	258	1	6.4	Asia
171	Timor-Leste	1	1	4	0.1	Asia
176	Turkey	51	22	7	1.4	Asia
177	Turkmenistan	19	71	32	2.2	Asia
181	United Arab Emirates	16	135	5	2.8	Asia
186	Uzbekistan	25	101	8	2.4	Asia
189	Vietnam	111	2	1	2.0	Asia
190	Yemen	6	0	0	0.1	Asia

In [124]:

1 asia.mean()

Out[124]:

```

beer_servings      37.045455
spirit_servings    60.840909
wine_servings      9.068182
total_litres_of_pure_alcohol  2.170455
dtype: float64

```

In [125]:

```
1 asia
```

Out[125]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
12	Bahrain	42	63	7	2.0	Asia
13	Bangladesh	0	0	0	0.0	Asia
19	Bhutan	23	0	0	0.4	Asia
24	Brunei	31	2	1	0.6	Asia
30	Cambodia	57	65	1	2.2	Asia
36	China	79	192	8	5.0	Asia
46	North Korea	0	0	0	0.0	Asia
77	India	9	114	0	2.2	Asia
78	Indonesia	5	1	0	0.1	Asia
79	Iran	0	0	0	0.0	Asia
80	Iraq	9	3	0	0.2	Asia
82	Israel	63	69	9	2.5	Asia
85	Japan	77	202	16	7.0	Asia
86	Jordan	6	21	1	0.5	Asia
87	Kazakhstan	124	246	12	6.8	Asia
90	Kuwait	0	0	0	0.0	Asia
91	Kyrgyzstan	31	97	6	2.4	Asia
92	Laos	62	0	123	6.2	Asia
94	Lebanon	20	55	31	1.9	Asia
102	Malaysia	13	4	0	0.3	Asia
103	Maldives	0	0	0	0.0	Asia
112	Mongolia	77	189	8	4.9	Asia
116	Myanmar	5	1	0	0.1	Asia
119	Nepal	5	6	0	0.2	Asia
127	Oman	22	16	1	0.7	Asia
128	Pakistan	0	0	0	0.0	Asia
134	Philippines	71	186	1	4.6	Asia
137	Qatar	1	42	7	0.9	Asia
138	South Korea	140	16	9	9.8	Asia
141	Russian Federation	247	326	73	11.5	Asia
149	Saudi Arabia	0	5	0	0.1	Asia
154	Singapore	60	12	11	1.5	Asia
161	Sri Lanka	16	104	0	2.2	Asia
167	Syria	5	35	16	1.0	Asia
168	Tajikistan	2	15	0	0.3	Asia
169	Thailand	99	258	1	6.4	Asia
171	Timor-Leste	1	1	4	0.1	Asia
176	Turkey	51	22	7	1.4	Asia
177	Turkmenistan	19	71	32	2.2	Asia
181	United Arab Emirates	16	135	5	2.8	Asia
186	Uzbekistan	25	101	8	2.4	Asia
189	Vietnam	111	2	1	2.0	Asia
190	Yemen	6	0	0	0.1	Asia

In [132]:

```
1 asia[asia['country'] == 'India']
```

Out[132]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
77	India	9	114	0	2.2	Asia

In []:

```
1
```


In [135]:

```
1 africa = df[df['continent'] == 'Africa']
```

In [136]:

1 africa

Out[136]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
2	Algeria	25	0	14	0.7	Africa
4	Angola	217	57	45	5.9	Africa
18	Benin	34	4	13	1.1	Africa
22	Botswana	173	35	35	5.4	Africa
26	Burkina Faso	25	7	7	4.3	Africa
27	Burundi	88	0	0	6.3	Africa
28	Cote d'Ivoire	37	1	7	4.0	Africa
29	Cabo Verde	144	56	16	4.0	Africa
31	Cameroon	147	1	4	5.8	Africa
33	Central African Republic	17	2	1	1.8	Africa
34	Chad	15	1	1	0.4	Africa
38	Comoros	1	3	1	0.1	Africa
39	Congo	76	1	9	1.7	Africa
47	DR Congo	32	3	1	2.3	Africa
49	Djibouti	15	44	3	1.1	Africa
53	Egypt	6	4	1	0.2	Africa
55	Equatorial Guinea	92	0	233	5.8	Africa
56	Eritrea	18	0	0	0.5	Africa
58	Ethiopia	20	3	0	0.7	Africa
62	Gabon	347	98	59	8.9	Africa
63	Gambia	8	0	1	2.4	Africa
66	Ghana	31	3	10	1.8	Africa
70	Guinea	9	0	2	0.2	Africa
71	Guinea-Bissau	28	31	21	2.5	Africa
88	Kenya	58	22	2	1.8	Africa
95	Lesotho	82	29	0	2.8	Africa
96	Liberia	19	152	2	3.1	Africa
97	Libya	0	0	0	0.0	Africa
100	Madagascar	26	15	4	0.8	Africa
101	Malawi	8	11	1	1.5	Africa
104	Mali	5	1	1	0.6	Africa
107	Mauritania	0	0	0	0.0	Africa
108	Mauritius	98	31	18	2.6	Africa
114	Morocco	12	6	10	0.5	Africa
115	Mozambique	47	18	5	1.3	Africa
117	Namibia	376	3	1	6.8	Africa
123	Niger	3	2	1	0.1	Africa
124	Nigeria	42	5	2	9.1	Africa
142	Rwanda	43	2	0	6.8	Africa
148	Sao Tome & Principe	56	38	140	4.2	Africa
150	Senegal	9	1	7	0.3	Africa
152	Seychelles	157	25	51	4.1	Africa
153	Sierra Leone	25	3	2	6.7	Africa
158	Somalia	0	0	0	0.0	Africa
159	South Africa	225	76	81	8.2	Africa
162	Sudan	8	13	0	1.7	Africa
164	Swaziland	90	2	2	4.7	Africa
172	Togo	36	2	19	1.3	Africa
175	Tunisia	51	3	20	1.3	Africa
179	Uganda	45	9	0	8.3	Africa
183	Tanzania	36	6	1	5.7	Africa
191	Zambia	32	19	4	2.5	Africa
192	Zimbabwe	64	18	4	4.7	Africa

In [137]:

```
1 africa.mean()
```

Out[137]:

```
beer_servings      61.471698
spirit_servings    16.339623
wine_servings      16.264151
total_litres_of_pure_alcohol  3.007547
dtype: float64
```

In []:

```
1
```