Week-2

July 3, 2020

*You are currently looking at **version 1.0** of this notebook. To download notebooks and datafiles, as well as get help on Jupyter notebooks in the Coursera platform, visit the Jupyter Notebook FAQ course resource.*

1 The Series Data Structure

[1]:
```
import pandas as pd
pd.Series?
```

[2]:
```
animals = ['Tiger', 'Bear', 'Moose']
pd.Series(animals)
```

[2]:
```
0    Tiger
1     Bear
2    Moose
dtype: object
```

[3]:
```
numbers = [1, 2, 3]
pd.Series(numbers)
```

[3]:
```
0    1
1    2
2    3
dtype: int64
```

[4]:
```
animals = ['Tiger', 'Bear', None]
pd.Series(animals)
```

[4]:
```
0    Tiger
1     Bear
2     None
dtype: object
```

[5]:
```
numbers = [1, 2, None]
pd.Series(numbers)
```

[5]:
```
0    1.0
1    2.0
```

```
2    NaN
dtype: float64
```

[6]:
```
import numpy as np
np.nan == None
```

[6]:
```
False
```

[7]:
```
np.nan == np.nan
```

[7]:
```
False
```

[8]:
```
np.isnan(np.nan)
```

[8]:
```
True
```

[9]:
```
sports = {'Archery': 'Bhutan',
          'Golf': 'Scotland',
          'Sumo': 'Japan',
          'Taekwondo': 'South Korea'}
s = pd.Series(sports)
s
```

[9]:
```
Archery           Bhutan
Golf            Scotland
Sumo               Japan
Taekwondo    South Korea
dtype: object
```

[10]:
```
s.index
```

[10]:
```
Index(['Archery', 'Golf', 'Sumo', 'Taekwondo'], dtype='object')
```

[11]:
```
s = pd.Series(['Tiger', 'Bear', 'Moose'], index=['India', 'America', 'Canada'])
s
```

[11]:
```
India      Tiger
America     Bear
Canada     Moose
dtype: object
```

[12]:
```
sports = {'Archery': 'Bhutan',
          'Golf': 'Scotland',
          'Sumo': 'Japan',
          'Taekwondo': 'South Korea'}
s = pd.Series(sports, index=['Golf', 'Sumo', 'Hockey'])
s
```

[12]:
```
Golf      Scotland
Sumo         Japan
Hockey         NaN
dtype: object
```

2 Querying a Series

[13]:
```
sports = {'Archery': 'Bhutan',
          'Golf': 'Scotland',
          'Sumo': 'Japan',
          'Taekwondo': 'South Korea'}
s = pd.Series(sports)
s
```

[13]:
```
Archery           Bhutan
Golf            Scotland
Sumo               Japan
Taekwondo    South Korea
dtype: object
```

[14]:
```
s.iloc[3]
```

[14]:
```
'South Korea'
```

[15]:
```
s.loc['Golf']
```

[15]:
```
'Scotland'
```

[16]:
```
s[3]
```

[16]:
```
'South Korea'
```

[17]:
```
s['Golf']
```

[17]:
```
'Scotland'
```

[18]:
```
sports = {99: 'Bhutan',
          100: 'Scotland',
          101: 'Japan',
          102: 'South Korea'}
s = pd.Series(sports)
```

[19]:
```
s[0] #This won't call s.iloc[0] as one might expect, it generates an error␣
 ↪instead
```

```
␣
↪---------------------------------------------------------------------------

KeyError                                  Traceback (most recent call␣
↪last)

<ipython-input-19-a5f43d492595> in <module>()
----> 1 s[0] #This won't call s.iloc[0] as one might expect, it generates an␣
↪error instead
```

```
      /opt/conda/lib/python3.6/site-packages/pandas/core/series.py in␣
↪__getitem__(self, key)
      601         key = com._apply_if_callable(key, self)
      602         try:
  --> 603             result = self.index.get_value(self, key)
      604
      605             if not is_scalar(result):


      /opt/conda/lib/python3.6/site-packages/pandas/indexes/base.py in␣
↪get_value(self, series, key)
     2167         try:
     2168             return self._engine.get_value(s, k,
   -> 2169                                           tz=getattr(series.dtype,␣
↪'tz', None))
     2170         except KeyError as e1:
     2171             if len(self) > 0 and self.inferred_type in ['integer',␣
↪'boolean']:


      pandas/index.pyx in pandas.index.IndexEngine.get_value (pandas/index.c:
↪3557)()


      pandas/index.pyx in pandas.index.IndexEngine.get_value (pandas/index.c:
↪3240)()


      pandas/index.pyx in pandas.index.IndexEngine.get_loc (pandas/index.c:
↪4279)()


      pandas/src/hashtable_class_helper.pxi in pandas.hashtable.Int64HashTable.
↪get_item (pandas/hashtable.c:8564)()


      pandas/src/hashtable_class_helper.pxi in pandas.hashtable.Int64HashTable.
↪get_item (pandas/hashtable.c:8508)()


      KeyError: 0
```

[20]:
```
s = pd.Series([100.00, 120.00, 101.00, 3.00])
s
```

[20]:
```
0    100.0
1    120.0
2    101.0
3      3.0
dtype: float64
```

[21]:
```
total = 0
for item in s:
    total+=item
print(total)
```

```
324.0
```

[22]:
```
import numpy as np

total = np.sum(s)
print(total)
```

```
324.0
```

[23]:
```
#this creates a big series of random numbers
s = pd.Series(np.random.randint(0,1000,10000))
s.head()
```

[23]:
```
0    555
1    147
2    789
3    349
4     40
dtype: int64
```

[24]:
```
len(s)
```

[24]:
```
10000
```

[25]:
```
%%timeit -n 100
summary = 0
for item in s:
    summary+=item
```

```
3.43 ms ś 368 ţs per loop (mean ś std. dev. of 7 runs, 100 loops each)
```

[26]:
```
%%timeit -n 100
summary = np.sum(s)
```

```
The slowest run took 9.04 times longer than the fastest. This could mean that an
intermediate result is being cached.
423 ţs ś 375 ţs per loop (mean ś std. dev. of 7 runs, 100 loops each)
```

[27]:
```
s+=2 #adds two to each item in s using broadcasting
s.head()
```

[27]:
```
0    557
1    149
2    791
3    351
4     42
dtype: int64
```

[28]:
```
for label, value in s.iteritems():
    s.set_value(label, value+2)
s.head()
```

[28]:
```
0    559
1    151
2    793
3    353
4     44
dtype: int64
```

[29]:
```
%%timeit -n 10
s = pd.Series(np.random.randint(0,1000,10000))
for label, value in s.iteritems():
    s.loc[label]= value+2
```

```
2.82 s ś 268 ms per loop (mean ś std. dev. of 7 runs, 10 loops each)
```

[30]:
```
%%timeit -n 10
s = pd.Series(np.random.randint(0,1000,10000))
s+=2
```

```
252 ţs ś 25 ţs per loop (mean ś std. dev. of 7 runs, 10 loops each)
```

[31]:
```
s = pd.Series([1, 2, 3])
s.loc['Animal'] = 'Bears'
s
```

[31]:
```
0            1
1            2
2            3
Animal    Bears
dtype: object
```

[32]:
```
original_sports = pd.Series({'Archery': 'Bhutan',
                             'Golf': 'Scotland',
                             'Sumo': 'Japan',
                             'Taekwondo': 'South Korea'})
cricket_loving_countries = pd.Series(['Australia',
                                      'Barbados',
```

```
                                        'Pakistan',
                                        'England'],
                                       index=['Cricket',
                                              'Cricket',
                                              'Cricket',
                                              'Cricket'])
all_countries = original_sports.append(cricket_loving_countries)
```

[33]:
```
original_sports
```

[33]:
```
Archery           Bhutan
Golf            Scotland
Sumo               Japan
Taekwondo    South Korea
dtype: object
```

[34]:
```
cricket_loving_countries
```

[34]:
```
Cricket    Australia
Cricket     Barbados
Cricket     Pakistan
Cricket      England
dtype: object
```

[35]:
```
all_countries
```

[35]:
```
Archery           Bhutan
Golf            Scotland
Sumo               Japan
Taekwondo    South Korea
Cricket        Australia
Cricket         Barbados
Cricket         Pakistan
Cricket          England
dtype: object
```

[36]:
```
all_countries.loc['Cricket']
```

[36]:
```
Cricket    Australia
Cricket     Barbados
Cricket     Pakistan
Cricket      England
dtype: object
```

3 The DataFrame Data Structure

[37]:
```
import pandas as pd
purchase_1 = pd.Series({'Name': 'Chris',
                        'Item Purchased': 'Dog Food',
                        'Cost': 22.50})
```

```
purchase_2 = pd.Series({'Name': 'Kevyn',
                        'Item Purchased': 'Kitty Litter',
                        'Cost': 2.50})
purchase_3 = pd.Series({'Name': 'Vinod',
                        'Item Purchased': 'Bird Seed',
                        'Cost': 5.00})
df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index=['Store 1',␣
 ↪'Store 1', 'Store 2'])
df.head()
```

[37]:

	Cost	Item Purchased	Name
Store 1	22.5	Dog Food	Chris
Store 1	2.5	Kitty Litter	Kevyn
Store 2	5.0	Bird Seed	Vinod

[38]:
```
df.loc['Store 2']
```

[38]:
```
Cost                       5
Item Purchased    Bird Seed
Name                  Vinod
Name: Store 2, dtype: object
```

[39]:
```
type(df.loc['Store 2'])
```

[39]:
```
pandas.core.series.Series
```

[40]:
```
df.loc['Store 1']
```

[40]:

	Cost	Item Purchased	Name
Store 1	22.5	Dog Food	Chris
Store 1	2.5	Kitty Litter	Kevyn

[41]:
```
df.loc['Store 1', 'Cost']
```

[41]:
```
Store 1    22.5
Store 1     2.5
Name: Cost, dtype: float64
```

[42]:
```
df.T
```

[42]:

	Store 1	Store 1	Store 2
Cost	22.5	2.5	5
Item Purchased	Dog Food	Kitty Litter	Bird Seed
Name	Chris	Kevyn	Vinod

[43]:
```
df.T.loc['Cost']
```

[43]:
```
Store 1    22.5
Store 1     2.5
Store 2       5
Name: Cost, dtype: object
```

[44]:
```
df['Cost']
```

[44]:
```
Store 1    22.5
Store 1     2.5
Store 2     5.0
Name: Cost, dtype: float64
```

[45]:
```
df.loc['Store 1']['Cost']
```

[45]:
```
Store 1    22.5
Store 1     2.5
Name: Cost, dtype: float64
```

[46]:
```
df.loc[:,['Name', 'Cost']]
```

[46]:

	Name	Cost
Store 1	Chris	22.5
Store 1	Kevyn	2.5
Store 2	Vinod	5.0

[47]:
```
df.drop('Store 1')
```

[47]:

	Cost	Item Purchased	Name
Store 2	5.0	Bird Seed	Vinod

[48]:
```
df
```

[48]:

	Cost	Item Purchased	Name
Store 1	22.5	Dog Food	Chris
Store 1	2.5	Kitty Litter	Kevyn
Store 2	5.0	Bird Seed	Vinod

[49]:
```
copy_df = df.copy()
copy_df = copy_df.drop('Store 1')
copy_df
```

[49]:

	Cost	Item Purchased	Name
Store 2	5.0	Bird Seed	Vinod

[50]:
```
copy_df.drop?
```

[51]:
```
del copy_df['Name']
copy_df
```

[51]:

	Cost	Item Purchased
Store 2	5.0	Bird Seed

[52]:
```
df['Location'] = None
df
```

[52]:

	Cost	Item Purchased	Name	Location
Store 1	22.5	Dog Food	Chris	None
Store 1	2.5	Kitty Litter	Kevyn	None
Store 2	5.0	Bird Seed	Vinod	None

4 Dataframe Indexing and Loading

[53]:
```
costs = df['Cost']
costs
```

[53]:
```
Store 1    22.5
Store 1     2.5
Store 2     5.0
Name: Cost, dtype: float64
```

[54]:
```
costs+=2
costs
```

[54]:
```
Store 1    24.5
Store 1     4.5
Store 2     7.0
Name: Cost, dtype: float64
```

df1 = pd.read_csv('')

[55]:
```
!cat olympics.csv
```

```
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
, Summer,01 !,02 !,03 !,Total, Winter,01 !,02 !,03 !,Total, Games,01 !,02
!,03 !,Combined total
Afghanistanã(AFG),13,0,0,2,2,0,0,0,0,0,13,0,0,2,2
Algeriaã(ALG),12,5,2,8,15,3,0,0,0,0,15,5,2,8,15
Argentinaã(ARG),23,18,24,28,70,18,0,0,0,0,41,18,24,28,70
Armeniaã(ARM),5,1,2,9,12,6,0,0,0,0,11,1,2,9,12
Australasiaã(ANZ) [ANZ],2,3,4,5,12,0,0,0,0,0,2,3,4,5,12
Australiaã(AUS) [AUS] [Z],25,139,152,177,468,18,5,3,4,12,43,144,155,181,480
Austriaã(AUT),26,18,33,35,86,22,59,78,81,218,48,77,111,116,304
Azerbaijanã(AZE),5,6,5,15,26,5,0,0,0,0,10,6,5,15,26
Bahamasã(BAH),15,5,2,5,12,0,0,0,0,0,15,5,2,5,12
Bahrainã(BRN),8,0,0,1,1,0,0,0,0,0,8,0,0,1,1
Barbadosã(BAR) [BAR],11,0,0,1,1,0,0,0,0,0,11,0,0,1,1
Belarusã(BLR),5,12,24,39,75,6,6,4,5,15,11,18,28,44,90
Belgiumã(BEL),25,37,52,53,142,20,1,1,3,5,45,38,53,56,147
Bermudaã(BER),17,0,0,1,1,7,0,0,0,0,24,0,0,1,1
Bohemiaã(BOH) [BOH] [Z],3,0,1,3,4,0,0,0,0,0,3,0,1,3,4
Botswanaã(BOT),9,0,1,0,1,0,0,0,0,0,9,0,1,0,1
Brazilã(BRA),21,23,30,55,108,7,0,0,0,0,28,23,30,55,108
British West Indiesã(BWI) [BWI],1,0,0,2,2,0,0,0,0,0,1,0,0,2,2
Bulgariaã(BUL) [H],19,51,85,78,214,19,1,2,3,6,38,52,87,81,220
Burundiã(BDI),5,1,0,0,1,0,0,0,0,0,5,1,0,0,1
Cameroonã(CMR),13,3,1,1,5,1,0,0,0,0,14,3,1,1,5
Canadaã(CAN),25,59,99,121,279,22,62,56,52,170,47,121,155,173,449
Chileã(CHI) [I],22,2,7,4,13,16,0,0,0,0,38,2,7,4,13
Chinaã(CHN) [CHN],9,201,146,126,473,10,12,22,19,53,19,213,168,145,526
Colombiaã(COL),18,2,6,11,19,1,0,0,0,0,19,2,6,11,19
```

Costa Ricaă(CRC),14,1,1,2,4,6,0,0,0,0,20,1,1,2,4
Ivory Coastă(CIV) [CIV],12,0,1,0,1,0,0,0,0,0,12,0,1,0,1
Croatiaă(CRO),6,6,7,10,23,7,4,6,1,11,13,10,13,11,34
Cubaă(CUB) [Z],19,72,67,70,209,0,0,0,0,0,19,72,67,70,209
Cyprusă(CYP),9,0,1,0,1,10,0,0,0,0,19,0,1,0,1
Czech Republicã(CZE) [CZE],5,14,15,15,44,6,7,9,8,24,11,21,24,23,68
Czechoslovakiaă(TCH) [TCH],16,49,49,45,143,16,2,8,15,25,32,51,57,60,168
Denmarkă(DEN) [Z],26,43,68,68,179,13,0,1,0,1,39,43,69,68,180
Djiboutiă(DJI) [B],7,0,0,1,1,0,0,0,0,0,7,0,0,1,1
Dominican Republicã(DOM),13,3,2,1,6,0,0,0,0,0,13,3,2,1,6
Ecuadoră(ECU),13,1,1,0,2,0,0,0,0,0,13,1,1,0,2
Egyptă(EGY) [EGY] [Z],21,7,9,10,26,1,0,0,0,0,22,7,9,10,26
Eritreaă(ERI),4,0,0,1,1,0,0,0,0,0,4,0,0,1,1
Estoniaă(EST),11,9,9,15,33,9,4,2,1,7,20,13,11,16,40
Ethiopiaă(ETH),12,21,7,17,45,2,0,0,0,0,14,21,7,17,45
Finlandă(FIN),24,101,84,117,302,22,42,62,57,161,46,143,146,174,463
Franceă(FRA) [O] [P] [Z],27,202,223,246,671,22,31,31,47,109,49,233,254,293,780
Gabonă(GAB),9,0,1,0,1,0,0,0,0,0,9,0,1,0,1
Georgiaă(GEO),5,6,5,14,25,6,0,0,0,0,11,6,5,14,25
Germanyă(GER) [GER] [Z],15,174,182,217,573,11,78,78,53,209,26,252,260,270,782
United Team of Germanyă(EUA) [EUA],3,28,54,36,118,3,8,6,5,19,6,36,60,41,137
East Germanyă(GDR) [GDR],5,153,129,127,409,6,39,36,35,110,11,192,165,162,519
West Germanyă(FRG) [FRG],5,56,67,81,204,6,11,15,13,39,11,67,82,94,243
Ghanaă(GHA) [GHA],13,0,1,3,4,1,0,0,0,0,14,0,1,3,4
Great Britainã(GBR) [GBR]
[Z],27,236,272,272,780,22,10,4,12,26,49,246,276,284,806
Greeceă(GRE) [Z],27,30,42,39,111,18,0,0,0,0,45,30,42,39,111
Grenadaă(GRN),8,1,0,0,1,0,0,0,0,0,8,1,0,0,1
Guatemalaă(GUA),13,0,1,0,1,1,0,0,0,0,14,0,1,0,1
Guyanaă(GUY) [GUY],16,0,0,1,1,0,0,0,0,0,16,0,0,1,1
Haitiă(HAI) [J],14,0,1,1,2,0,0,0,0,0,14,0,1,1,2
Hong Kongă(HKG) [HKG],15,1,1,1,3,4,0,0,0,0,19,1,1,1,3
Hungaryă(HUN),25,167,144,165,476,22,0,2,4,6,47,167,146,169,482
Icelandă(ISL),19,0,2,2,4,17,0,0,0,0,36,0,2,2,4
Indiaă(IND) [F],23,9,6,11,26,9,0,0,0,0,32,9,6,11,26
Indonesiaă(INA),14,6,10,11,27,0,0,0,0,0,14,6,10,11,27
Iranã(IRI) [K],15,15,20,25,60,10,0,0,0,0,25,15,20,25,60
Iraqă(IRQ),13,0,0,1,1,0,0,0,0,0,13,0,0,1,1
Irelandă(IRL),20,9,8,12,29,6,0,0,0,0,26,9,8,12,29
Israelă(ISR),15,1,1,5,7,6,0,0,0,0,21,1,1,5,7
Italyă(ITA) [M] [S],26,198,166,185,549,22,37,34,43,114,48,235,200,228,663
Jamaicaă(JAM) [JAM],16,17,30,20,67,7,0,0,0,0,23,17,30,20,67
Japanã(JPN),21,130,126,142,398,20,10,17,18,45,41,140,143,160,443
Kazakhstanã(KAZ),5,16,17,19,52,6,1,3,3,7,11,17,20,22,59
Kenyaă(KEN),13,25,32,29,86,3,0,0,0,0,16,25,32,29,86
North Koreaă(PRK),9,14,12,21,47,8,0,1,1,2,17,14,13,22,49
South Koreaă(KOR),16,81,82,80,243,17,26,17,10,53,33,107,99,90,296
Kuwaitã(KUW),12,0,0,2,2,0,0,0,0,0,12,0,0,2,2

Kyrgyzstană(KGZ),5,0,1,2,3,6,0,0,0,0,11,0,1,2,3
Latviaă(LAT),10,3,11,5,19,10,0,4,3,7,20,3,15,8,26
Lebanonă(LIB),16,0,2,2,4,16,0,0,0,0,32,0,2,2,4
Liechtensteină(LIE),16,0,0,0,0,18,2,2,5,9,34,2,2,5,9
Lithuaniaă(LTU),8,6,5,10,21,8,0,0,0,0,16,6,5,10,21
Luxembourgă(LUX) [O],22,1,1,0,2,8,0,2,0,2,30,1,3,0,4
Macedoniaă(MKD),5,0,0,1,1,5,0,0,0,0,10,0,0,1,1
Malaysiaă(MAS) [MAS],12,0,3,3,6,0,0,0,0,0,12,0,3,3,6
Mauritiusă(MRI),8,0,0,1,1,0,0,0,0,0,8,0,0,1,1
Mexicoă(MEX),22,13,21,28,62,8,0,0,0,0,30,13,21,28,62
Moldovaă(MDA),5,0,2,5,7,6,0,0,0,0,11,0,2,5,7
Mongoliaă(MGL),12,2,9,13,24,13,0,0,0,0,25,2,9,13,24
Montenegroă(MNE),2,0,1,0,1,2,0,0,0,0,4,0,1,0,1
Moroccoă(MAR),13,6,5,11,22,6,0,0,0,0,19,6,5,11,22
Mozambiqueă(MOZ),9,1,0,1,2,0,0,0,0,0,9,1,0,1,2
Namibiaă(NAM),6,0,4,0,4,0,0,0,0,0,6,0,4,0,4
Netherlandsă(NED) [Z],25,77,85,104,266,20,37,38,35,110,45,114,123,139,376
Netherlands Antillesă(AHO) [AHO] [I],13,0,1,0,1,2,0,0,0,0,15,0,1,0,1
New Zealandă(NZL) [NZL],22,42,18,39,99,15,0,1,0,1,37,42,19,39,100
Nigeră(NIG),11,0,0,1,1,0,0,0,0,0,11,0,0,1,1
Nigeriaă(NGR),15,3,8,12,23,0,0,0,0,0,15,3,8,12,23
Norwayă(NOR) [Q],24,56,49,43,148,22,118,111,100,329,46,174,160,143,477
Pakistană(PAK),16,3,3,4,10,2,0,0,0,0,18,3,3,4,10
Panamaă(PAN),16,1,0,2,3,0,0,0,0,0,16,1,0,2,3
Paraguayă(PAR),11,0,1,0,1,1,0,0,0,0,12,0,1,0,1
Peruă(PER) [L],17,1,3,0,4,2,0,0,0,0,19,1,3,0,4
Philippinesă(PHI),20,0,2,7,9,4,0,0,0,0,24,0,2,7,9
Polandă(POL),20,64,82,125,271,22,6,7,7,20,42,70,89,132,291
Portugală(POR),23,4,8,11,23,7,0,0,0,0,30,4,8,11,23
Puerto Ricoă(PUR),17,0,2,6,8,6,0,0,0,0,23,0,2,6,8
Qatară(QAT),8,0,0,4,4,0,0,0,0,0,8,0,0,4,4
Romaniaă(ROU),20,88,94,119,301,20,0,0,1,1,40,88,94,120,302
Russiaă(RUS) [RUS],5,132,121,142,395,6,49,40,35,124,11,181,161,177,519
Russian Empireă(RU1) [RU1],3,1,4,3,8,0,0,0,0,0,3,1,4,3,8
Soviet Unionă(URS) [URS],9,395,319,296,1010,9,78,57,59,194,18,473,376,355,1204
Unified Teamă(EUN) [EUN],1,45,38,29,112,1,9,6,8,23,2,54,44,37,135
Saudi Arabiaă(KSA),10,0,1,2,3,0,0,0,0,0,10,0,1,2,3
Senegală(SEN),13,0,1,0,1,5,0,0,0,0,18,0,1,0,1
Serbiaă(SRB) [SRB],3,1,2,4,7,2,0,0,0,0,5,1,2,4,7
Serbia and Montenegroă(SCG) [SCG],3,2,4,3,9,3,0,0,0,0,6,2,4,3,9
Singaporeă(SIN),15,0,2,2,4,0,0,0,0,0,15,0,2,2,4
Slovakiaă(SVK) [SVK],5,7,9,8,24,6,2,2,1,5,11,9,11,9,29
Sloveniaă(SLO),6,4,6,9,19,7,2,4,9,15,13,6,10,18,34
South Africaă(RSA),18,23,26,27,76,6,0,0,0,0,24,23,26,27,76
Spaină(ESP) [Z],22,37,59,35,131,19,1,0,1,2,41,38,59,36,133
Sri Lankaă(SRI) [SRI],16,0,2,0,2,0,0,0,0,0,16,0,2,0,2
Sudană(SUD),11,0,1,0,1,0,0,0,0,0,11,0,1,0,1
Surinameă(SUR) [E],11,1,0,1,2,0,0,0,0,0,11,1,0,1,2

```
Swedenă(SWE) [Z],26,143,164,176,483,22,50,40,54,144,48,193,204,230,627
Switzerlandă(SUI),27,47,73,65,185,22,50,40,48,138,49,97,113,113,323
Syriaă(SYR),12,1,1,1,3,0,0,0,0,0,12,1,1,1,3
Chinese Taipeiă(TPE) [TPE] [TPE2],13,2,7,12,21,11,0,0,0,0,24,2,7,12,21
Tajikistană(TJK),5,0,1,2,3,4,0,0,0,0,9,0,1,2,3
Tanzaniaă(TAN) [TAN],12,0,2,0,2,0,0,0,0,0,12,0,2,0,2
Thailandă(THA),15,7,6,11,24,3,0,0,0,0,18,7,6,11,24
Togoă(TOG),9,0,0,1,1,1,0,0,0,0,10,0,0,1,1
Tongaă(TGA),8,0,1,0,1,1,0,0,0,0,9,0,1,0,1
Trinidad and Tobagoă(TRI) [TRI],16,2,5,11,18,3,0,0,0,0,19,2,5,11,18
Tunisiaă(TUN),13,3,3,4,10,0,0,0,0,0,13,3,3,4,10
Turkeyă(TUR),21,39,25,24,88,16,0,0,0,0,37,39,25,24,88
Ugandaă(UGA),14,2,3,2,7,0,0,0,0,0,14,2,3,2,7
Ukraineă(UKR),5,33,27,55,115,6,2,1,4,7,11,35,28,59,122
United Arab Emiratesă(UAE),8,1,0,0,1,0,0,0,0,0,8,1,0,0,1
United Statesă(USA) [P] [Q] [R]
[Z],26,976,757,666,2399,22,96,102,84,282,48,1072,859,750,2681
Uruguayă(URU),20,2,2,6,10,1,0,0,0,0,21,2,2,6,10
Uzbekistană(UZB),5,5,5,10,20,6,1,0,0,1,11,6,5,10,21
Venezuelaă(VEN),17,2,2,8,12,4,0,0,0,0,21,2,2,8,12
Vietnamă(VIE),14,0,2,0,2,0,0,0,0,0,14,0,2,0,2
Virgin Islandsă(ISV),11,0,1,0,1,7,0,0,0,0,18,0,1,0,1
Yugoslaviaă(YUG) [YUG],16,26,29,28,83,14,0,3,1,4,30,26,32,29,87
Independent Olympic Participantsă(IOP) [IOP],1,0,1,2,3,0,0,0,0,0,1,0,1,2,3
Zambiaă(ZAM) [ZAM],12,0,1,1,2,0,0,0,0,0,12,0,1,1,2
Zimbabweă(ZIM) [ZIM],12,3,4,1,8,1,0,0,0,0,13,3,4,1,8
Mixed teamă(ZZX) [ZZX],3,8,5,4,17,0,0,0,0,0,3,8,5,4,17
Totals,27,4809,4775,5130,14714,22,959,958,948,2865,49,5768,5733,6078,17579
```

[56]:
```
import pandas as pd
df = pd.read_csv('olympics.csv')
df.head()
```

[56]:

	0	1	2	3	4	5	6	7	8
0	NaN	Summer	01 !	02 !	03 !	Total	Winter	01 !	02 !
1	Afghanistană(AFG)	13	0	0	2	2	0	0	0
2	Algeriaă(ALG)	12	5	2	8	15	3	0	0
3	Argentinaă(ARG)	23	18	24	28	70	18	0	0
4	Armeniaă(ARM)	5	1	2	9	12	6	0	0

	9	10	11	12	13	14	15
0	03 !	Total	Games	01 !	02 !	03 !	Combined total
1	0	0	13	0	0	2	2
2	0	0	15	5	2	8	15
3	0	0	41	18	24	28	70
4	0	0	11	1	2	9	12

[57]:
```
df = pd.read_csv('olympics.csv', index_col=0, skiprows=1)
df.head()
```

[57]:
```
                          Summer  01 !  02 !  03 !  Total   Winter  01 !.1  \
Afghanistană(AFG)             13     0     0     2      2         0       0
Algeriaă(ALG)                 12     5     2     8     15         3       0
Argentinaă(ARG)               23    18    24    28     70        18       0
Armeniaă(ARM)                  5     1     2     9     12         6       0
Australasiaă(ANZ) [ANZ]        2     3     4     5     12         0       0

                          02 !.1  03 !.1  Total.1   Games  01 !.2  02 !.2  \
Afghanistană(AFG)              0       0        0      13       0       0
Algeriaă(ALG)                  0       0        0      15       5       2
Argentinaă(ARG)                0       0        0      41      18      24
Armeniaă(ARM)                  0       0        0      11       1       2
Australasiaă(ANZ) [ANZ]        0       0        0       2       3       4

                          03 !.2  Combined total
Afghanistană(AFG)              2               2
Algeriaă(ALG)                  8              15
Argentinaă(ARG)               28              70
Armeniaă(ARM)                  9              12
Australasiaă(ANZ) [ANZ]        5              12
```

[58]:
```
for col in df.columns:
    if col[:2] == '01':
        df.rename(columns={col: 'Gold'+col[4:]}, inplace=True)
    if col[:2] == '02':
        df.rename(columns={col: 'Silver'+col[4:]}, inplace=True)
    if col[:2] == '03':
        df.rename(columns={col: 'Bronze'+col[4:]}, inplace=True)
    if col[:3] == 'Man':
        df.rename(columns={col:  '#'+col[4:]}, inplace=True)
df.head()
```

[58]:
```
                          Summer  Gold  Silver  Bronze  Total   Winter  \
Afghanistană(AFG)             13     0       0       2      2         0
Algeriaă(ALG)                 12     5       2       8     15         3
Argentinaă(ARG)               23    18      24      28     70        18
Armeniaă(ARM)                  5     1       2       9     12         6
Australasiaă(ANZ) [ANZ]        2     3       4       5     12         0

                          Gold.1  Silver.1  Bronze.1  Total.1   Games  Gold.2  \
Afghanistană(AFG)              0         0         0        0      13       0
Algeriaă(ALG)                  0         0         0        0      15       5
Argentinaă(ARG)                0         0         0        0      41      18
Armeniaă(ARM)                  0         0         0        0      11       1
```

```
Australasiaă(ANZ) [ANZ]       0         0         0       0       2       3

                         Silver.2  Bronze.2  Combined total
Afghanistană(AFG)               0         2               2
Algeriaă(ALG)                   2         8              15
Argentinaă(ARG)                24        28              70
Armeniaă(ARM)                   2         9              12
Australasiaă(ANZ) [ANZ]         4         5              12
```

[59]:
```
df['Gold'] > 0
```

[59]:
```
Afghanistană(AFG)                          False
Algeriaă(ALG)                               True
Argentinaă(ARG)                             True
Armeniaă(ARM)                               True
Australasiaă(ANZ) [ANZ]                     True
Australiaă(AUS) [AUS] [Z]                   True
Austriaă(AUT)                               True
Azerbaijană(AZE)                            True
Bahamasă(BAH)                               True
Bahrainǎ(BRN)                              False
Barbadosă(BAR) [BAR]                       False
Belarusă(BLR)                               True
Belgiumă(BEL)                               True
Bermudaă(BER)                              False
Bohemiaă(BOH) [BOH] [Z]                    False
Botswanaă(BOT)                             False
Brazilă(BRA)                                True
British West Indiesă(BWI) [BWI]            False
Bulgariaă(BUL) [H]                          True
Burundiă(BDI)                               True
Cameroonă(CMR)                              True
Canadaă(CAN)                                True
Chileă(CHI) [I]                             True
Chinaă(CHN) [CHN]                           True
Colombiaă(COL)                              True
Costa Ricaă(CRC)                            True
Ivory Coastă(CIV) [CIV]                    False
Croatiaă(CRO)                               True
Cubaă(CUB) [Z]                              True
Cyprusă(CYP)                               False
                                           ...
Sri Lankaă(SRI) [SRI]                      False
Sudanǎ(SUD)                                False
Surinameă(SUR) [E]                          True
Swedenă(SWE) [Z]                            True
Switzerlandă(SUI)                           True
Syriaă(SYR)                                 True
```

```
Chinese Taipeiă(TPE) [TPE] [TPE2]                True
Tajikistană(TJK)                                False
Tanzaniaă(TAN) [TAN]                            False
Thailandă(THA)                                   True
Togoă(TOG)                                      False
Tongaă(TGA)                                     False
Trinidad and Tobagoă(TRI) [TRI]                  True
Tunisiaă(TUN)                                    True
Turkeyă(TUR)                                     True
Ugandaă(UGA)                                     True
Ukraineă(UKR)                                    True
United Arab Emiratesă(UAE)                       True
United Statesă(USA) [P] [Q] [R] [Z]              True
Uruguayă(URU)                                    True
Uzbekistană(UZB)                                 True
Venezuelaă(VEN)                                  True
Vietnamă(VIE)                                   False
Virgin Islandsă(ISV)                            False
Yugoslaviaă(YUG) [YUG]                           True
Independent Olympic Participantsă(IOP) [IOP]    False
Zambiaă(ZAM) [ZAM]                              False
Zimbabweă(ZIM) [ZIM]                             True
Mixed teamă(ZZX) [ZZX]                           True
Totals                                           True
Name: Gold, dtype: bool
```

[63]:
```
d1 = df.where(df['Gold']> 0)
d1.head()
```

[63]:

	Summer	Gold	Silver	Bronze	Total	Winter	\
Afghanistană(AFG)	NaN	NaN	NaN	NaN	NaN	NaN	
Algeriaă(ALG)	12.0	5.0	2.0	8.0	15.0	3.0	
Argentinaă(ARG)	23.0	18.0	24.0	28.0	70.0	18.0	
Armeniaă(ARM)	5.0	1.0	2.0	9.0	12.0	6.0	
Australasiaă(ANZ) [ANZ]	2.0	3.0	4.0	5.0	12.0	0.0	

	Gold.1	Silver.1	Bronze.1	Total.1	Games	Gold.2	\
Afghanistană(AFG)	NaN	NaN	NaN	NaN	NaN	NaN	
Algeriaă(ALG)	0.0	0.0	0.0	0.0	15.0	5.0	
Argentinaă(ARG)	0.0	0.0	0.0	0.0	41.0	18.0	
Armeniaă(ARM)	0.0	0.0	0.0	0.0	11.0	1.0	
Australasiaă(ANZ) [ANZ]	0.0	0.0	0.0	0.0	2.0	3.0	

	Silver.2	Bronze.2	Combined total
Afghanistană(AFG)	NaN	NaN	NaN
Algeriaă(ALG)	2.0	8.0	15.0
Argentinaă(ARG)	24.0	28.0	70.0
Armeniaă(ARM)	2.0	9.0	12.0

```
Australasiaă(ANZ) [ANZ]       4.0       5.0            12.0
```

[64]:
```
df[(df['Gold'] == 0) & (df['Gold.1'] > 0)]
```

[64]:
```
                      Summer  Gold  Silver  Bronze  Total   Winter  Gold.1  \
Liechtensteină(LIE)       16     0       0       0      0       18       2

                     Silver.1  Bronze.1  Total.1   Games  Gold.2  Silver.2  \
Liechtensteină(LIE)         2         5        9       34       2         2

                     Bronze.2  Combined total
Liechtensteină(LIE)         5               9
```

[65]:
```
df = pd.read_csv('olympics.csv', index_col=0, skiprows=1)
for col in df.columns:
    if col[:2] == '01':
        df.rename(columns={col: 'Gold'+col[4:]}, inplace=True)
    if col[:2] == '02':
        df.rename(columns={col: 'Silver'+col[4:]}, inplace=True)
    if col[:2] == '03':
        df.rename(columns={col: 'Bronze'+col[4:]}, inplace=True)

df['Country'] = df.index
df = df.set_index('Gold')
df['Gold'] = df.index
df = df.set_index('Country')
df = df.reset_index()
df.head()
```

[65]:
```
                   Country   Summer  Silver  Bronze  Total   Winter  Gold.1  \
0        Afghanistană(AFG)       13       0       2      2        0       0
1            Algeriaă(ALG)       12       2       8     15        3       0
2          Argentinaă(ARG)       23      24      28     70       18       0
3            Armeniaă(ARM)        5       2       9     12        6       0
4  Australasiaă(ANZ) [ANZ]        2       4       5     12        0       0

   Silver.1  Bronze.1  Total.1   Games  Gold.2  Silver.2  Bronze.2  \
0         0         0        0      13       0         0         2
1         0         0        0      15       5         2         8
2         0         0        0      41      18        24        28
3         0         0        0      11       1         2         9
4         0         0        0       2       3         4         5

   Combined total  Gold
0               2     0
1              15     5
2              70    18
3              12     1
4              12     3
```

5 Missing values

[66]:
```
import pandas as pd
dd = pd.read_csv('census.csv')
dd = dd[dd['SUMLEV'] == 50]
dd = dd.set_index(['STNAME', 'CTYNAME'])
dd.loc[ [('Alabama', 'Autauga County'), ('Alabama', 'Blount')] ]
```

[66]:
```
                        SUMLEV  REGION  DIVISION  STATE  COUNTY  \
STNAME  CTYNAME
Alabama Autauga County    50.0     3.0       6.0    1.0     1.0
        Blount             NaN     NaN       NaN    NaN     NaN

                        CENSUS2010POP  ESTIMATESBASE2010  POPESTIMATE2010  \
STNAME  CTYNAME
Alabama Autauga County        54571.0            54571.0          54660.0
        Blount                    NaN                NaN              NaN

                        POPESTIMATE2011  POPESTIMATE2012     ...       \
STNAME  CTYNAME                                              ...
Alabama Autauga County          55253.0          55175.0     ...
        Blount                      NaN              NaN     ...

                        RDOMESTICMIG2011  RDOMESTICMIG2012  RDOMESTICMIG2013  \
STNAME  CTYNAME
Alabama Autauga County          7.242091         -2.915927         -3.012349
        Blount                       NaN               NaN               NaN

                        RDOMESTICMIG2014  RDOMESTICMIG2015  RNETMIG2011  \
STNAME  CTYNAME
Alabama Autauga County          2.265971         -2.530799     7.606016
        Blount                       NaN               NaN          NaN

                        RNETMIG2012  RNETMIG2013  RNETMIG2014  RNETMIG2015
STNAME  CTYNAME
Alabama Autauga County    -2.626146    -2.722002      2.59227    -2.187333
        Blount                  NaN          NaN          NaN          NaN

[2 rows x 98 columns]
```

[67]:
```
import pandas as pd
df = pd.read_csv('log.csv')
df = df.set_index(['time', 'user', 'video'])
df = df.reset_index()

df.head()
```

[67]:
```
         time    user       video  playback position paused  volume
0  1469974424  cheryl  intro.html                  5  False    10.0
1  1469974454  cheryl  intro.html                  6    NaN     NaN
2  1469974544  cheryl  intro.html                  9    NaN     NaN
3  1469974574  cheryl  intro.html                 10    NaN     NaN
4  1469977514     bob  intro.html                  1    NaN     NaN
```

[68]:
```
dd.loc['Alabama', 'Barbour County']
dd.loc[ [('Alabama', 'Barbour County'), ('Alabama', 'Baldwin County')] ]
```

[68]:
```
                         SUMLEV  REGION  DIVISION  STATE  COUNTY  \
STNAME  CTYNAME
Alabama Barbour County      50       3         6      1       5
        Baldwin County      50       3         6      1       3

                         CENSUS2010POP  ESTIMATESBASE2010  POPESTIMATE2010  \
STNAME  CTYNAME
Alabama Barbour County           27457              27457            27341
        Baldwin County          182265             182265           183193

                         POPESTIMATE2011  POPESTIMATE2012     ...       \
STNAME  CTYNAME                                              ...
Alabama Barbour County             27226            27159     ...
        Baldwin County            186659           190396     ...

                         RDOMESTICMIG2011  RDOMESTICMIG2012  RDOMESTICMIG2013  \
STNAME  CTYNAME
Alabama Barbour County         -4.728132         -2.500690         -7.056824
        Baldwin County         14.832960         17.647293         21.845705

                         RDOMESTICMIG2014  RDOMESTICMIG2015  RNETMIG2011  \
STNAME  CTYNAME
Alabama Barbour County         -3.904217        -10.543299    -4.874741
        Baldwin County         19.243287         17.197872    15.844176

                         RNETMIG2012  RNETMIG2013  RNETMIG2014  RNETMIG2015
STNAME  CTYNAME
Alabama Barbour County     -2.758113    -7.167664    -3.978583   -10.543299
        Baldwin County     18.559627    22.727626    20.317142    18.293499

[2 rows x 98 columns]
```

[69]:
```
df = df.reset_index()
df = df.set_index(['time', 'user'])
df
```

[69]:
```
                    index          video  playback position paused  volume
time       user
1469974424 cheryl       0     intro.html                  5  False    10.0
```

```
1469974454 cheryl      1     intro.html                  6    NaN     NaN
1469974544 cheryl      2     intro.html                  9    NaN     NaN
1469974574 cheryl      3     intro.html                 10    NaN     NaN
1469977514 bob         4     intro.html                  1    NaN     NaN
1469977544 bob         5     intro.html                  1    NaN     NaN
1469977574 bob         6     intro.html                  1    NaN     NaN
1469977604 bob         7     intro.html                  1    NaN     NaN
1469974604 cheryl      8     intro.html                 11    NaN     NaN
1469974694 cheryl      9     intro.html                 14    NaN     NaN
1469974724 cheryl     10     intro.html                 15    NaN     NaN
1469974454 sue        11  advanced.html                 24    NaN     NaN
1469974524 sue        12  advanced.html                 25    NaN     NaN
1469974424 sue        13  advanced.html                 23  False    10.0
1469974554 sue        14  advanced.html                 26    NaN     NaN
1469974624 sue        15  advanced.html                 27    NaN     NaN
1469974654 sue        16  advanced.html                 28    NaN     5.0
1469974724 sue        17  advanced.html                 29    NaN     NaN
1469974484 cheryl     18     intro.html                  7    NaN     NaN
1469974514 cheryl     19     intro.html                  8    NaN     NaN
1469974754 sue        20  advanced.html                 30    NaN     NaN
1469974824 sue        21  advanced.html                 31    NaN     NaN
1469974854 sue        22  advanced.html                 32    NaN     NaN
1469974924 sue        23  advanced.html                 33    NaN     NaN
1469977424 bob        24     intro.html                  1   True    10.0
1469977454 bob        25     intro.html                  1    NaN     NaN
1469977484 bob        26     intro.html                  1    NaN     NaN
1469977634 bob        27     intro.html                  1    NaN     NaN
1469977664 bob        28     intro.html                  1    NaN     NaN
1469974634 cheryl     29     intro.html                 12    NaN     NaN
1469974664 cheryl     30     intro.html                 13    NaN     NaN
1469977694 bob        31     intro.html                  1    NaN     NaN
1469977724 bob        32     intro.html                  1    NaN     NaN
```

[ ]:
```
df = df.fillna(method='ffill')
df.head()
```