

Ch05_Data Gathering and Cleaning

September 5, 2018

1 Chapter 5: Data Gathering and Cleaning

```
In [46]: import numpy as np
         np.random.randn(5, 3)
```

```
Out[46]: array([[ -0.61061723,  0.72415256,  1.66399174],
                [-1.7389832 ,  0.85504316,  1.01567421],
                [ 0.61267705,  0.50718253, -1.9840606 ],
                [-0.48915204,  0.82164722, -0.37554022],
                [ 1.72835338,  0.13632558, -2.83155864]])
```

```
In [47]: import pandas as pd
         import numpy as np
```

```
dataset = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'], columns=['stock1', 'stock2', 'stock3'])
dataset.rename(columns={"one": 'stock1', "two": 'stock2', "three": 'stock3'}, inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
```

```
print (dataset)
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	NaN	NaN	NaN
c	-0.641868	1.272449	0.256177
d	NaN	NaN	NaN
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	NaN	NaN	NaN
h	0.165526	1.315884	0.192126

```
In [48]: print (dataset['stock1'].isnull())
```

a	False
b	True
c	False
d	True

```
e    False
f    False
g     True
h    False
Name: stock1, dtype: bool
```

```
In [49]: print (dataset)
         dataset.fillna(0)
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	NaN	NaN	NaN
c	-0.641868	1.272449	0.256177
d	NaN	NaN	NaN
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	NaN	NaN	NaN
h	0.165526	1.315884	0.192126

```
Out[49]:
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	0.000000	0.000000	0.000000
c	-0.641868	1.272449	0.256177
d	0.000000	0.000000	0.000000
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	0.000000	0.000000	0.000000
h	0.165526	1.315884	0.192126

```
In [50]: # Fill missing values forward
         print (dataset)
         dataset.fillna(method='pad')
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	NaN	NaN	NaN
c	-0.641868	1.272449	0.256177
d	NaN	NaN	NaN
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	NaN	NaN	NaN
h	0.165526	1.315884	0.192126

```
Out[50]:
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	0.673643	0.736574	-0.075297

c	-0.641868	1.272449	0.256177
d	-0.641868	1.272449	0.256177
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	0.470721	1.568048	0.108818
h	0.165526	1.315884	0.192126

```
In [51]: print (dataset)
         dataset.dropna()
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	NaN	NaN	NaN
c	-0.641868	1.272449	0.256177
d	NaN	NaN	NaN
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	NaN	NaN	NaN
h	0.165526	1.315884	0.192126

```
Out[51]:
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
c	-0.641868	1.272449	0.256177
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
h	0.165526	1.315884	0.192126

```
In [52]: print (dataset)
         dataset.replace(np.nan, 0 )
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	NaN	NaN	NaN
c	-0.641868	1.272449	0.256177
d	NaN	NaN	NaN
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	NaN	NaN	NaN
h	0.165526	1.315884	0.192126

```
Out[52]:
```

	stock1	stock2	stock3
a	0.673643	0.736574	-0.075297
b	0.000000	0.000000	0.000000
c	-0.641868	1.272449	0.256177
d	0.000000	0.000000	0.000000
e	-0.767121	0.716501	0.675500
f	0.470721	1.568048	0.108818
g	0.000000	0.000000	0.000000
h	0.165526	1.315884	0.192126

2 Read CSV files

```
In [53]: import pandas as pd
         sales = pd.read_csv("Sales.csv")
         print ("\n\n<<<<<<< First 5 records <<<<<<<\n\n" )
         print (sales.head())
```

<<<<<<< First 5 records <<<<<<<

	SALES_ID	SALES_BY_REGION	JANUARY	FEBRUARY	MARCH	APRIL	\
0	1	AUH	3,469.00	n.a.	not available	3,642.00	
1	1	SHJ	5,840.00	5,270.00	4,114.00	5,605.00	
2	1	-1	2,967.00	2,425.00	5,353.00	n.a.	
3	2	AUH	1,328.00	-1	1,574.00	2,343.00	
4	3	SHJ	2,473.00	1,421.00	3,606.00	1,314.00	

	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	\
0	5,803.00	5,662.00	1,896.00	2,293.00	2,583.00	5,233.00	4,421.00	
1	4,387.00	5,026.00	4,055.00	2,782.00	4,578.00	4,993.00	2,859.00	
2	5,027.00	4,078.00	3,858.00	1,927.00	3,527.00	4,179.00	1,571.00	
3	3,826.00	4,932.00	1,710.00	3,221.00	3,381.00	1,313.00	1,765.00	
4	1,413.00	2,091.00	3,270.00	3,346.00	2,080.00	1,539.00	2,630.00	

	DECEMBER
0	4,071.00
1	4,853.00
2	5,551.00
3	1,214.00
4	1,697.00

```
In [54]: print ("\n\n<<<<<<< Last 5 records <<<<<<<\n\n" )
         print (sales.tail())
```

<<<<<<< Last 5 records <<<<<<<

	SALES_ID	SALES_BY_REGION	JANUARY	FEBRUARY	MARCH	APRIL	\
93	29	FUJ	2,492.00	1,833.00	2,982.00	4,292.00	
94	30	AJM	2,832.00	5,978.00	1,684.00	1,550.00	
95	30	FUJ	3,402.00	5,283.00	2,229.00	3,758.00	
96	30	AJM	2,028.00	2,006.00	5,120.00	5,959.00	
97	30	FUJ	5,549.00	1,302.00	1,929.00	2,822.00	

	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	\
93	5,540.00	5,847.00	909	2,339.00	4,868.00	5,207.00	5,938.00	
94	1,194.00	3,737.00	5,779.00	4,441.00	1,213.00	3,711.00	5,384.00	
95	1,427.00	1,057.00	5,277.00	5,231.00	3,909.00	4,345.00	5,287.00	
96	3,127.00	3,962.00	4,780.00	3,200.00	1,836.00	2,623.00	1,607.00	
97	5,379.00	1,243.00	3,075.00	4,358.00	5,106.00	2,322.00	2,409.00	
DECEMBER								
93	1,793.00							
94	1,293.00							
95	2,638.00							
96	2,371.00							
97	1,069.00							

```
In [55]: #import pandas as pd
salesNrows = pd.read_csv("Sales.csv", nrows=4)
salesNrows
```

```
Out [55]:
```

	SALES_ID	SALES_BY_REGION	JANUARY	FEBRUARY	MARCH	APRIL	\
0	1	AUH	3,469.00	n.a.	not available	3,642.00	
1	1	SHJ	5,840.00	5,270.00	4,114.00	5,605.00	
2	1	-1	2,967.00	2,425.00	5,353.00	n.a.	
3	2	AUH	1,328.00	-1	1,574.00	2,343.00	

	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	\
0	5,803.00	5,662.00	1,896.00	2,293.00	2,583.00	5,233.00	4,421.00	
1	4,387.00	5,026.00	4,055.00	2,782.00	4,578.00	4,993.00	2,859.00	
2	5,027.00	4,078.00	3,858.00	1,927.00	3,527.00	4,179.00	1,571.00	
3	3,826.00	4,932.00	1,710.00	3,221.00	3,381.00	1,313.00	1,765.00	
DECEMBER								
0	4,071.00							
1	4,853.00							
2	5,551.00							
3	1,214.00							

```
In [56]: salesNrows.rename(columns={"SALES_ID": 'ID', "SALES_BY_REGION": 'REGION'}, inplace=True)
salesNrows
```

```
Out [56]:
```

	ID	REGION	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	\
0	1	AUH	3,469.00	n.a.	not available	3,642.00	5,803.00	5,662.00	
1	1	SHJ	5,840.00	5,270.00	4,114.00	5,605.00	4,387.00	5,026.00	
2	1	-1	2,967.00	2,425.00	5,353.00	n.a.	5,027.00	4,078.00	
3	2	AUH	1,328.00	-1	1,574.00	2,343.00	3,826.00	4,932.00	
			JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER	
0	1		1,896.00	2,293.00	2,583.00	5,233.00	4,421.00	4,071.00	
1	1		4,055.00	2,782.00	4,578.00	4,993.00	2,859.00	4,853.00	

```

2  3,858.00  1,927.00  3,527.00  4,179.00  1,571.00  5,551.00
3  1,710.00  3,221.00  3,381.00  1,313.00  1,765.00  1,214.00

```

3 Find unique values

```

In [57]: print (len(salesNrows['JANUARY'].unique()))
         print (len(salesNrows['REGION'].unique()))
         print (salesNrows['JANUARY'].unique())

```

4

3

```
['3,469.00' '5,840.00' '2,967.00' '1,328.00']
```

```

In [58]: #[0, 1, 2] or ['SALES_ID' , 'SALES_BY_REGION', 'JANUARY']
         salesNrows = pd.read_csv("Sales.csv", nrows=4, usecols=[0, 1, 6])
         salesNrows

```

```

Out[58]:  SALES_ID  SALES_BY_REGION      MAY
0         1           AUH  5,803.00
1         1           SHJ  4,387.00
2         1           -1  5,027.00
3         2           AUH  3,826.00

```

```

In [60]: # Read specific fields of data [0, 1, 2] or
         #[ 'SALES_ID' , 'SALES_BY_REGION', 'JANUARY']
         salesNrows = pd.read_csv("Sales.csv", nrows=4,
                                   usecols=['SALES_ID' , 'SALES_BY_REGION', 'FEBRUARY', 'MARCH'])
         salesNrows

```

```

Out[60]:  SALES_ID  SALES_BY_REGION  FEBRUARY      MARCH
0         1           AUH      n.a.  not available
1         1           SHJ  5,270.00    4,114.00
2         1           -1  2,425.00    5,353.00
3         2           AUH      -1    1,574.00

```

```

In [61]: sales = pd.read_csv("Sales.csv", nrows=7,
                             na_values=["n.a.", "not available"])
         mydata = sales.head(7)
         mydata

```

```

Out[61]:  SALES_ID  SALES_BY_REGION  JANUARY  FEBRUARY      MARCH      APRIL      MAY  \
0         1           AUH  3,469.00      NaN      NaN  3,642.00  5,803.00
1         1           SHJ  5,840.00  5,270.00  4,114.00  5,605.00  4,387.00
2         1           -1  2,967.00  2,425.00  5,353.00      NaN  5,027.00
3         2           AUH  1,328.00      -1  1,574.00  2,343.00  3,826.00
4         3           SHJ  2,473.00  1,421.00  3,606.00  1,314.00  1,413.00
5         3           NaN      NaN      956  1,297.00  1,984.00  2,744.00

```

			AUH	2,634.00	2,143.00	3,698.00	5,767.00	2,782.00
		JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
0	5,662.00	1,896.00	2,293.00	2,583.00	5,233.00	4,421.00	4,071.00	
1	5,026.00	4,055.00	2,782.00	4,578.00	4,993.00	2,859.00	4,853.00	
2	4,078.00	3,858.00	1,927.00	3,527.00	4,179.00	1,571.00	5,551.00	
3	4,932.00	1,710.00	3,221.00	3,381.00	1,313.00	1,765.00	1,214.00	
4	2,091.00	3,270.00	3,346.00	2,080.00	1,539.00	2,630.00	1,697.00	
5	5,793.00	2,261.00	5,607.00	2,437.00	4,328.00	3,317.00	5,390.00	
6	4,444.00	5,036.00	4,805.00	5,792.00	5,256.00	4,096.00	3,170.00	

```
In [62]: sales = pd.read_csv("Sales.csv", nrows=7,
na_values=["n.a.", "not available", -1])
mydata = sales.head(7)
mydata
```

```
Out[62]:
```

	SALES_ID	SALES_BY_REGION	JANUARY	FEBRUARY	MARCH	APRIL	MAY	\
0	1	AUH	3,469.00	NaN	NaN	3,642.00	5,803.00	
1	1	SHJ	5,840.00	5,270.00	4,114.00	5,605.00	4,387.00	
2	1	NaN	2,967.00	2,425.00	5,353.00	NaN	5,027.00	
3	2	AUH	1,328.00	NaN	1,574.00	2,343.00	3,826.00	
4	3	SHJ	2,473.00	1,421.00	3,606.00	1,314.00	1,413.00	
5	3	NaN	NaN	956	1,297.00	1,984.00	2,744.00	
6	3	AUH	2,634.00	2,143.00	3,698.00	5,767.00	2,782.00	

		JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
0	5,662.00	1,896.00	2,293.00	2,583.00	5,233.00	4,421.00	4,071.00	
1	5,026.00	4,055.00	2,782.00	4,578.00	4,993.00	2,859.00	4,853.00	
2	4,078.00	3,858.00	1,927.00	3,527.00	4,179.00	1,571.00	5,551.00	
3	4,932.00	1,710.00	3,221.00	3,381.00	1,313.00	1,765.00	1,214.00	
4	2,091.00	3,270.00	3,346.00	2,080.00	1,539.00	2,630.00	1,697.00	
5	5,793.00	2,261.00	5,607.00	2,437.00	4,328.00	3,317.00	5,390.00	
6	4,444.00	5,036.00	4,805.00	5,792.00	5,256.00	4,096.00	3,170.00	

4 Data Integration

4.1 Read Data

```
In [63]: import pandas as pd
```

```
a = pd.read_csv("1. Export1_Columns.csv")
b = pd.read_csv("1. Export2_Columns.csv")
```

```
In [64]: a.head()
```

```
Out[64]:
```

	Country Name	Country Code	2004	2005	2006	2007
0	Benin	BEN	811	940	869	1076
1	Burkina Faso	BFA	548	532	673	714

2	Bangladesh	BGD	7257	9995	11745	13530
3	Bulgaria	BGR	10713	12703	16151	23263
4	Bahrain	BHR	10337	13397	15662	17314

In [65]: b.head()

```
Out[65]:
```

	Country Name	Country Code	2008	2009	2010	2011	2012	2013	2014
0	Benin	BEN	1312	1039	991	1040	1154	1518	1656
1	Burkina Faso	BFA	834	1063	1727	2681	2849	3166	3551
2	Bangladesh	BGD	16181	17360	18472	25627	26887	29305	34344
3	Bulgaria	BGR	28591	21964	26836	35488	33975	37260	37845
4	Bahrain	BHR	21231	15705	17880	22945	22853	0	0

In [66]: a.head()

```
Out[66]:
```

	Country Name	Country Code	2004	2005	2006	2007
0	Benin	BEN	811	940	869	1076
1	Burkina Faso	BFA	548	532	673	714
2	Bangladesh	BGD	7257	9995	11745	13530
3	Bulgaria	BGR	10713	12703	16151	23263
4	Bahrain	BHR	10337	13397	15662	17314

```
In [67]: b.drop('2014', axis=1, inplace=True)
columns = ['2013', '2012']
b.drop(columns, inplace=True, axis=1)
b.drop(b.columns[[3]], axis=1, inplace=True)
b.head()
```

```
Out[67]:
```

	Country Name	Country Code	2008	2010	2011
0	Benin	BEN	1312	991	1040
1	Burkina Faso	BFA	834	1727	2681
2	Bangladesh	BGD	16181	18472	25627
3	Bulgaria	BGR	28591	26836	35488
4	Bahrain	BHR	21231	17880	22945

```
In [68]: mergedDataSet = a.merge(b, on="Country Name")
mergedDataSet.head()
```

```
Out[68]:
```

	Country Name	Country Code_x	2004	2005	2006	2007	Country Code_y \
0	Benin	BEN	811	940	869	1076	BEN
1	Burkina Faso	BFA	548	532	673	714	BFA
2	Bangladesh	BGD	7257	9995	11745	13530	BGD
3	Bulgaria	BGR	10713	12703	16151	23263	BGR
4	Bahrain	BHR	10337	13397	15662	17314	BHR

	2008	2010	2011
0	1312	991	1040
1	834	1727	2681
2	16181	18472	25627
3	28591	26836	35488
4	21231	17880	22945


```
In [69]: dataX = a.merge(b)
dataX.head()
```

```
Out[69]:
```

	Country Name	Country Code	2004	2005	2006	2007	2008	2010	2011
0	Benin	BEN	811	940	869	1076	1312	991	1040
1	Burkina Faso	BFA	548	532	673	714	834	1727	2681
2	Bangladesh	BGD	7257	9995	11745	13530	16181	18472	25627
3	Bulgaria	BGR	10713	12703	16151	23263	28591	26836	35488
4	Bahrain	BHR	10337	13397	15662	17314	21231	17880	22945

5 Merge two data sets using Index

5.0.1 Rows Union

```
In [70]: Data1 = a.head()
Data1=Data1.reset_index()
Data1
```

```
Out[70]:
```

	index	Country Name	Country Code	2004	2005	2006	2007
0	0	Benin	BEN	811	940	869	1076
1	1	Burkina Faso	BFA	548	532	673	714
2	2	Bangladesh	BGD	7257	9995	11745	13530
3	3	Bulgaria	BGR	10713	12703	16151	23263
4	4	Bahrain	BHR	10337	13397	15662	17314

```
In [71]: Data2 = a.tail()
Data2=Data2.reset_index()
Data2
```

```
Out[71]:
```

	index	Country Name	Country Code	2004	2005	2006	2007
0	228	Yemen, Rep.	YEM	5048	6852	7873	0
1	229	South Africa	ZAF	58216	68172	79519	93339
2	230	Congo, Dem. Rep.	COD	2341	2442	2765	6540
3	231	Zambia	ZMB	2087	2550	4158	4722
4	232	Zimbabwe	ZWE	2001	1931	1957	2000

```
In [72]: # stack the DataFrames on top of each other
VerticalStack = pd.concat((Data1, Data2), axis=0)
VerticalStack
```

```
Out[72]:
```

	index	Country Name	Country Code	2004	2005	2006	2007
0	0	Benin	BEN	811	940	869	1076
1	1	Burkina Faso	BFA	548	532	673	714
2	2	Bangladesh	BGD	7257	9995	11745	13530
3	3	Bulgaria	BGR	10713	12703	16151	23263
4	4	Bahrain	BHR	10337	13397	15662	17314
0	228	Yemen, Rep.	YEM	5048	6852	7873	0
1	229	South Africa	ZAF	58216	68172	79519	93339
2	230	Congo, Dem. Rep.	COD	2341	2442	2765	6540
3	231	Zambia	ZMB	2087	2550	4158	4722
4	232	Zimbabwe	ZWE	2001	1931	1957	2000

6 Read Jason data

```
In [73]: import json
        data = '''{
            "name" : "Ossama",
            "phone" : {
                "type" : "intl",
                "number" : "+971 50 244 5467"
            },
            "email" : {
                "hide" : "No"
            }
        }'''
        info = json.loads(data)
        print ('Name:',info["name"])
        print ('Hide:',info["email"]["hide"])
```

Name: Ossama

Hide: No

```
In [74]: input = '''[
            { "id" : "001",
              "x" : "5",
              "name" : "Ossama"
            } ,
            { "id" : "009",
              "x" : "10",
              "name" : "Omar"
            }
        ]'''
        info = json.loads(input)
        print ('User count:', len(info))
        for item in info:
            print ('\nName', item['name'])
            print ('Id', item['id'])
            print ('Attribute', item['x'])
```

User count: 2

Name Ossama

Id 001

Attribute 5

Name Omar

Id 009

Attribute 10

6.1 Read Jason from the cloud

```
In [91]: import urllib.request
import json

with urllib.request.urlopen("http://python-data.dr-chuck.net/comments_244984.json") as
    uh = url.read()

print ('Retrieving', url)

data = uh
print ('Retrieved',len(data),'characters')

try:
    js = json.loads(str(data))
except:
    js = None

print (json.dumps(js, indent=4))
```

```
Retrieving <http.client.HTTPResponse object at 0x7fccc9b41470>
Retrieved 2722 characters
null
```

```
In [99]: from urllib.request import urlopen
import json
req = urlopen("http://python-data.dr-chuck.net/comments_244984.json")
json = json.loads(req.read())
print (json)
print (json['comments'])
```

```
{'note': 'This file contains the actual data for your assignment', 'comments': [{'name': 'Abaan',
[{'name': 'Abaan', 'count': 98}, {'name': 'Ashna', 'count': 95}, {'name': 'Dante', 'count': 94},
```

```
In [100]: sum=0
counter=0
for i in range(len(json["comments"])):
    counter+=1
    Name = json["comments"][i]["name"]
    Count = json["comments"][i]["count"]
    sum+=int(Count)
    print (Name," ", Count)

print ("\nCount: ", counter)
print ("Sum: ", sum)
```

Abaan	98
Ashna	95
Dante	94
Isabel	93
Fearne	92
Kriss	91
Janani	87
Karhys	85
Megg	84
Luisa	83
Thorben	79
Kaelan	77
Ceirín	75
Lileidh	70
Angelika	70
Amelka	69
Justin	69
Muneeb	68
Antoine	64
Ivar	61
Kaid	60
Dakotah	58
Nadeem	58
Marybeth	55
Ashlyn	55
Kaydin	50
Obieluem	48
Cairn	46
Ala	45
Vithujan	38
Ivory	34
Rosalyn	33
Kaywan	32
Pedro	31
Bharath	30
Eshaal	29
Aliya	28
Sephíroth	27
Minah	25
Murdo	22
Ata	21
Remonae	17
Muskaan	17
Lottie	17
Giane	9
Dineo	6
Zoe	5
Raul	4

Tammylee 2
Morna 1

Count: 50
Sum: 2507

```
In [101]: import json
          with open('comments.json') as json_data:
              jasondta = json.load(json_data)
              print(jasondta)
```

```
{'note': 'This file contains the actual data for your assignment', 'comments': [{'name': 'Abaan'
```

```
In [102]: sum=0
          counter=0
          for i in range(len(jasondta["comments"])):
              counter+=1
              Name = jasondta["comments"][i]["name"]
              Count = jasondta["comments"][i]["count"]
              sum+=int(Count)
              print (Name, " ", Count)

          print ("\nCount: ", counter)
          print ("Sum: ", sum)
```

Abaan 98
Ashna 95
Dante 94
Isabel 93
Fearne 92
Kriss 91
Janani 87
Karhys 85
Megg 84
Luisa 83
Thorben 79
Kaelan 77
Ceirin 75
Lileidh 70
Angelika 70
Amelka 69
Justin 69
Muneeb 68
Antoine 64
Ivar 61
Kaid 60
Dakotah 58

Nadeem	58
Marybeth	55
Ashlyn	55
Kaydin	50
Obieluem	48
Cairn	46
Ala	45
Vithujan	38
Ivory	34
Rosalyn	33
Kaywan	32
Pedro	31
Bharath	30
Eshaal	29
Aliya	28
Sephiroth	27
Minah	25
Murdo	22
Ata	21
Remonae	17
Muskaan	17
Lottie	17
Giane	9
Dineo	6
Zoe	5
Raul	4
Tammylee	2
Morna	1
Count:	50
Sum:	2507

7 Read and process HTML tags

```
In [103]: import urllib.request
          with urllib.request.urlopen("http://python-data.dr-chuck.net/known_by_Rona.html") as url:
              strhtml = url.read()
              #I'm guessing this would output the html source code?
              print(strhtml[:700])
```

```
b'<html>\n<head>\n<title>People that Rona knows</title>\n<style>\n.overlay{\n    opacity:0.99;\n
```

```
In [104]: import urllib
          from bs4 import BeautifulSoup

          response = urllib.request.urlopen('http://python-data.dr-chuck.net/known_by_Rona.html')
```

```

html_doc = response.read()

soup = BeautifulSoup(html_doc, 'html.parser')

print(html_doc[:700])
print("\n")
print(soup.title)
print(soup.title.string)
print(soup.a.string)

b'<html>\n<head>\n<title>People that Rona knows</title>\n<style>\n.overlay{\n    opacity:0.99;\n

<title>People that Rona knows</title>
People that Rona knows
Konar

In [106]: for x in soup.find_all('b'):
           print(x.string)

In [107]: import urllib
           from bs4 import BeautifulSoup

           response = urllib.request.urlopen('http://python-data.dr-chuck.net/known_by_Rona.html')
           html_doc = response.read()
           print(html_doc[:300])
           soup = BeautifulSoup(html_doc, 'html.parser')

           print("\n")
           counter=0
           for link in soup.findAll("a"):
               print(link.get("href"))
               if counter<10:
                   counter+=1
                   continue
               else: break

b'<html>\n<head>\n<title>People that Rona knows</title>\n<style>\n.overlay{\n    opacity:0.99;\n

http://python-data.dr-chuck.net/known_by_Konar.html
http://python-data.dr-chuck.net/known_by_Mohamad.html
http://python-data.dr-chuck.net/known_by_Keyra.html
http://python-data.dr-chuck.net/known_by_Jaxson.html
http://python-data.dr-chuck.net/known_by_Jordyn.html
http://python-data.dr-chuck.net/known_by_Cairn.html
http://python-data.dr-chuck.net/known_by_Bodhan.html
http://python-data.dr-chuck.net/known_by_Arianna.html

```

```
http://python-data.dr-chuck.net/known_by_Kiarrah.html
http://python-data.dr-chuck.net/known_by_Alannah.html
http://python-data.dr-chuck.net/known_by_Keira.html
```

```
In [108]: htmldata="""<html>
    <head>
    <title>
    The Dormouse's story
    </title>
</head>
<body>
<p class="title">
    <b>
    The Dormouse's story
    </b>
</p>
<p class="story">
    Once upon a time there were three little sisters; and their names were
    <a class="sister" href="http://example.com/elsie" id="link1">
    Elsie
    </a>
    ,
    <a class="sister" href="http://example.com/lacie" id="link2">
    Lacie
    </a>
    and
    <a class="sister" href="http://example.com/tillie" id="link2">
    Tillie
    </a>
    ; and they lived at the bottom of a well.
</p>
<p class="story">
    ...
</p>
</body>
</html>
"""

from bs4 import BeautifulSoup
soup = BeautifulSoup(htmldata, 'html.parser')
print(soup.prettify())
```

```
<html>
<head>
<title>
  The Dormouse's story
</title>
```



```

</head>
<body>
  <p class="title">
    <b>
      The Dormouse's story
    </b>
  </p>
  <p class="story">
    Once upon a time there were three little sisters; and their names were
    <a class="sister" href="http://example.com/elsie" id="link1">
      Elsie
    </a>
    ,
    <a class="sister" href="http://example.com/lacie" id="link2">
      Lacie
    </a>
    and
    <a class="sister" href="http://example.com/tillie" id="link2">
      Tillie
    </a>
    ; and they lived at the bottom of a well.
  </p>
  <p class="story">
    ...
  </p>
</body>
</html>

```

```
In [109]: soup.title
```

```
Out[109]: <title>
          The Dormouse's story
          </title>
```

```
In [110]: soup.title.name
```

```
Out[110]: 'title'
```

```
In [111]: soup.title.string
```

```
Out[111]: "\n    The Dormouse's story\n    "
```

```
In [112]: soup.title.parent.name
```

```
Out[112]: 'head'
```

```
In [113]: soup.p
```

```

Out[113]: <p class="title">
          <b>
              The Dormouse's story
          </b>
        </p>

In [114]: soup.p['class']

Out[114]: ['title']

In [115]: soup.a

Out[115]: <a class="sister" href="http://example.com/elsie" id="link1">
          Elsie
        </a>

In [116]: soup.find_all('a')

Out[116]: [<a class="sister" href="http://example.com/elsie" id="link1">
          Elsie
        </a>, <a class="sister" href="http://example.com/lacie" id="link2">
          Lacie
        </a>, <a class="sister" href="http://example.com/tillie" id="link2">
          Tillie
        </a>]

In [117]: soup.find(id="link2")

Out[117]: <a class="sister" href="http://example.com/lacie" id="link2">
          Lacie
        </a>

In [118]: for link in soup.find_all('a'):
          print(link.get('href'))

http://example.com/elsie
http://example.com/lacie
http://example.com/tillie

In [119]: print(soup.get_text())

```

The Dormouse's story

The Dormouse's story

Once upon a time there were three little sisters; and their names were

Elsie

,

Lacie

and

Tillie

; and they lived at the bottom of a well.

...

```
In [120]: htmldata="""<html>
    <head>
    <title>
    Python Book Verion 2018
    </title>
    </head>
    <body>
    <p class="title">
    <b>
    Author Name: Ossama Embarak
    </b>
    </p>
    <p class="story">
    Python techniques for gathering and cleaning data
    <a class="sister"
    href="https://leanpub.com/AgilePythonProgrammingAppliedForEveryone"
    id="link1">
    Data Cleaning
    </a>
```

```

        , Data Processing and Visulization
        <a class="sister" href="http://www.lulu.com/shop/ossama-

embarak/agile-python-programming-applied-for-

everyone/paperback/product-23694020.html" id="link2">
        Data Visualization
        </a>

    </p>
    <p class="story">
        @July 2018
    </p>
</body>
</html>
"""

from bs4 import BeautifulSoup
soup = BeautifulSoup(htmldata, 'html.parser')
print(soup.prettify())

<html>
<head>
<title>
    Python Book Verion 2018
</title>
</head>
<body>
    <p class="title">
        <b>
            Author Name: Ossama Embarak
        </b>
    </p>
    <p class="story">
        Python techniques for gathering and cleaning data
        <a class="sister" href="https://leanpub.com/AgilePythonProgrammingAppliedForEveryone" id="lin
            Data Cleaning
        </a>
        , Data Processing and Visulization
        <a class="sister" href="http://www.lulu.com/shop/ossama-

embarak/agile-python-programming-applied-for-

everyone/paperback/product-23694020.html" id="link2">
            Data Visualization
        </a>
    </p>
    <p class="story">

```

```
@July 2018
</p>
</body>
</html>
```

```
In [121]: print(soup.get_text())
```

Python Book Verion 2018

Author Name: Ossama Embarak

Python techniques for gathering and cleaning data

Data Cleaning

, Data Processing and Visulization

Data Visualization

@July 2018

```
In [128]: xmldata = """
    <?xml version="1.0"?>
    <data>
        <student name="Omar">
            <rank>2</rank>
            <year>2017</year>
            <GPA>3.5</GPA>
            <concentration name="Networking" Semester="7"/>
        </student>
```

```

        <student name="Ali">
            <rank>3</rank>
            <year>2016</year>
            <GPA>2.8</GPA>
            <concentration name="Security" Semester="6"/>
        </student>
        <student name="Osama">
            <rank>1</rank>
            <year>2018</year>
            <GPA>3.7</GPA>
            <concentration name="App Development" Semester="8"/>
        </student>
    </data>
"""
.strip()

```

```

In [129]: from xml.etree import ElementTree as ET
stuff = ET.fromstring(xmldata)
lst = stuff.findall('student')

print ('Students count:', len(lst))
for item in lst:
    print ("\nName:", item.get("name"))
    print ('concentration:', item.find("concentration").get("name"))
    print ('Rank:', item.find('rank').text)
    print ('GPA:', item.find("GPA").text)

```

Students count: 3

Name: Omar
concentration: Networking
Rank: 2
GPA: 3.5

Name: Ali
concentration: Security
Rank: 3
GPA: 2.8

Name: Osama
concentration: App Development
Rank: 1
GPA: 3.7

```

In [131]: value = ET.fromstring(xmldata).find('response/result/value')
if value:
    print ('Found value:', value.text)

```