Introduction to Pandas

Author: Srushti Shimpi ¶

LinkedIn Profile (https://www.linkedin.com/in/srushti-shimpi77/)

Importing the libraries

In [73]:
```
import pandas as pd
import numpy as np
```

Manipulating Text(String) data using Pandas

In [74]:
```
data1 = [['Apple',10],[' Orange',12],['Pear ',13],[1234,4],['Strwabery',28], [
'4567', 9],[np.nan, 11], ['pine@pple',5]]
df1 = pd.DataFrame(data1,columns=['Fruit','Count'], dtype = int)
df1
```

Out[74]:

	Fruit	Count
0	Apple	10
1	Orange	12
2	Pear	13
3	1234	4
4	Strwabery	28
5	4567	9
6	NaN	11
7	pine@pple	5

In [75]:
```
#Converts strings in the Series/Index to lower case.
#In result, it excluded 1234, because it wasn't a string. It has a numeric dat
a type
df1.Fruit.str.lower()
```

Out[75]:
```
0        apple
1       orange
2        pear
3          NaN
4    strwabery
5         4567
6          NaN
7    pine@pple
Name: Fruit, dtype: object
```

In [76]:
```
#Converts strings in the Series/Index to upper case. It works similary to lowe
r()
df1.Fruit.str.upper()
```

Out[76]:
```
0        APPLE
1       ORANGE
2        PEAR
3          NaN
4    STRWABERY
5         4567
6          NaN
7    PINE@PPLE
Name: Fruit, dtype: object
```

In [77]:
```
#Computes String length().
df1.Fruit.str.len()
```

Out[77]:
```
0    5.0
1    7.0
2    5.0
3    NaN
4    9.0
5    4.0
6    NaN
7    9.0
Name: Fruit, dtype: float64
```

In [78]:
```
#Show strip whitespace(including newline) from each string in the Series/index
from both the sides.
df1.Fruit.str.strip()
```

Out[78]:
```
0        Apple
1       Orange
2         Pear
3          NaN
4    Strwabery
5         4567
6          NaN
7    pine@pple
Name: Fruit, dtype: object
```

In [79]:
```
#Splits each string with the given pattern.
df1.Fruit.str.split(' ')
```

Out[79]:
```
0        [Apple]
1     [, Orange]
2       [Pear, ]
3            NaN
4    [Strwabery]
5         [4567]
6            NaN
7    [pine@pple]
Name: Fruit, dtype: object
```

In [80]:
```
data2 = [['Apple',10],[' Orange',12],['Pear ',13],['Strwabery',28], ['A4567',
9],[np.nan, 11], ['pine@pple',5]]
df2 = pd.DataFrame(data2,columns=['Fruit','Count'], dtype = int)
df2
```

Out[80]:

	Fruit	Count
0	Apple	10
1	Orange	12
2	Pear	13
3	Strwabery	28
4	A4567	9
5	NaN	11
6	pine@pple	5

In [81]:
```
#Concatenates the series/index elements with given separator.
df2.Fruit.str.cat(sep=', ')
```

Out[81]:
```
'Apple,  Orange, Pear , Strwabery, A4567, pine@pple'
```

In [82]:
```
#Returns the DataFrame with Encoded values.
df2.Fruit.str.get_dummies()
```

Out[82]:

	Orange	A4567	Apple	Pear	Strwabery	pine@pple
0	0	0	1	0	0	0
1	1	0	0	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	0	1	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	1

In [83]:
```
#Returns a Boolean value True for each element if the substring contains in th
e element, else False.
df2.Fruit.str.contains(' ')
```

Out[83]:
```
0    False
1     True
2     True
3    False
4    False
5      NaN
6    False
Name: Fruit, dtype: object
```

In [84]:
```
#Replaces the value a with the value b.
df2.Fruit.str.replace('@','a')
```

Out[84]:
```
0        Apple
1       Orange
2        Pear
3    Strwabery
4        A4567
5          NaN
6    pineapple
Name: Fruit, dtype: object
```

In [85]:
```
#Repeats each element with given number of times.
df2.Fruit.str.repeat(2)
```

Out[85]:
```
0            AppleApple
1         Orange Orange
2            Pear Pear
3    StrwaberyStrwabery
4            A4567A4567
5                   NaN
6    pine@pplepine@pple
Name: Fruit, dtype: object
```

In [86]:
```
#Gives number of appearance of pattern in each element.
df2.Fruit.str.count('e')
```

Out[86]:
```
0    1.0
1    1.0
2    1.0
3    1.0
4    0.0
5    NaN
6    2.0
Name: Fruit, dtype: float64
```

In [87]:
```
#Shows true if the element in the Series/Index starts with the pattern.
df2.Fruit.str.startswith ('A')
```

Out[87]:
```
0     True
1    False
2    False
3    False
4     True
5      NaN
6    False
Name: Fruit, dtype: object
```

In [88]:
```
#Shows true if the element in the Series/Index ends with the pattern.
df2.Fruit.str.endswith('e')
```

Out[88]:
```
0     True
1     True
2    False
3    False
4    False
5      NaN
6     True
Name: Fruit, dtype: object
```

In [89]:
```
#Gives first position of the first occurrence of the pattern.
df2.Fruit.str.find('e')
```

Out[89]:
```
0    4.0
1    6.0
2    1.0
3    6.0
4   -1.0
5    NaN
6    3.0
Name: Fruit, dtype: float64
```

In [90]:
```
#Shows all occurrence of the given pattern.
df2.Fruit.str.findall('e')
```

Out[90]:
```
0       [e]
1       [e]
2       [e]
3       [e]
4        []
5       NaN
6    [e, e]
Name: Fruit, dtype: object
```

In [91]:
```
#Swaps the case lower to upper, upper to lower case.
df2.Fruit.str.swapcase()
```

Out[91]:
```
0        aPPLE
1       oRANGE
2        pEAR
3    sTRWABERY
4        a4567
5          NaN
6    PINE@PPLE
Name: Fruit, dtype: object
```

In [92]:
```
#Checks whether all characters in each string in the Series/Index in lower cas
e or not.
df2.Fruit.str.islower()
```

Out[92]:
```
0    False
1    False
2    False
3    False
4    False
5      NaN
6     True
Name: Fruit, dtype: object
```

In [93]:
```
#Checks whether all characters in each string in the Series/Index in upper cas
e or not.
df2.Fruit.str.isupper()
```

Out[93]:
```
0    False
1    False
2    False
3    False
4     True
5      NaN
6    False
Name: Fruit, dtype: object
```

In [94]:
```
#Checks whether all characters in each string in the Series/Index are numeric.
df2.Fruit.str.isnumeric()
```

Out[94]:
```
0    False
1    False
2    False
3    False
4    False
5      NaN
6    False
Name: Fruit, dtype: object
```