9.1) Working with Text

Vitor Kamada

January 2020

Reference

Tables, Graphics, and Figures from

Principles and Techniques of Data Science

Lau et al. (2019): Ch 8 Working with Text

`https://www.textbook.ds100.org/ch/08/text_intro.html`

String Methods

Method	Description
`str[x:y]`	Slices `str`, returning indices x (inclusive) to y (not inclusive)
`str.lower()`	Returns a copy of a string with all letters converted to lowercase
`str.replace(a, b)`	Replaces all instances of the substring `a` in `str` with the substring `b`
`str.split(a)`	Returns substrings of `str` split at a substring `a`
`str.strip()`	Removes leading and trailing whitespace from `str`

`state`

	County	State
0	De Witt County	IL
1	Lac qui Parle County	MN
2	Lewis and Clark County	MT
3	St John the Baptist Parish	LA

`population`

	County	Population
0	DeWitt	16,798
1	Lac Qui Parle	8,067
2	Lewis & Clark	55,716
3	St. John the Baptist	43,044

Uniformize State

```
state['County'] = (state['County']
 .str.lower()
 .str.strip()
 .str.replace(' parish', '')
 .str.replace(' county', '')
 .str.replace('&', 'and')
 .str.replace('.', '')
 .str.replace(' ', '')
)
```

	County	State
0	De Witt County	IL
1	Lac qui Parle County	MN
2	Lewis and Clark County	MT
3	St John the Baptist Parish	LA

	County	State
0	dewitt	IL
1	lacquiparle	MN
2	lewisandclark	MT
3	stjohnthebaptist	LA

Uniformize Population

```
population['County'] = (population['County']
 .str.lower()
 .str.strip()
 .str.replace(' parish', '')
 .str.replace(' county', '')
 .str.replace('&', 'and')
 .str.replace('.', '')
 .str.replace(' ', '')
)
```

	County	Population
0	DeWitt	16,798
1	Lac Qui Parle	8,067
2	Lewis & Clark	55,716
3	St. John the Baptist	43,044

	County	Population
0	dewitt	16,798
1	lacquiparle	8,067
2	lewisandclark	55,716
3	stjohnthebaptist	43,044

```
state.merge(population, on='County')
```

	County	State	Population
0	dewitt	IL	16,798
1	lacquiparle	MN	8,067
2	lewisandclark	MT	55,716
3	stjohnthebaptist	LA	43,044

Regular Expressions

Description	Bracket Form	Shorthand
Alphanumeric character	`[a-zA-Z0-9]`	`\w`
Not an alphanumeric character	`[^a-zA-Z0-9]`	`\W`
Digit	`[0-9]`	`\d`
Not a digit	`[^0-9]`	`\D`
Whitespace	`[\t\n\f\r\p{Z}]`	`\s`
Not whitespace	`[^\t\n\f\r\p{z}]`	`\S`

```
import re
gmail_re = r'[a-zA-Z0-9]+@gmail\.com'
text = '''
From: email1@gmail.com
To: email2@yahoo.com and email3@gmail.com
'''
re.findall(gmail_re, text)
```

['email1@gmail.com', 'email3@gmail.com']

Meta Characters

Char	Description	Example	Matches	Doesn't Match
.	Any character except \n	`...`	abc	ab abcd
[ ]	Any character inside brackets	`[cb.]ar`	car .ar	jar
[^]	Any character *not* inside brackets	`[^b]ar`	car par	bar ar
*	≥ 0 or more of last symbol	`[pb]*ark`	bbark ark	dark
+	≥ 1 or more of last symbol	`[pb]+ark`	bbpark bark	dark ark
?	0 or 1 of last symbol	`s?he`	she he	the
{*n*}	Exactly *n* of last symbol	`hello{3}`	hellooo	hello
|	Pattern before or after bar	`we|[ui]s`	we us is	e s
\	Escapes next character	`\[hi\]`	[hi]	hi
^	Beginning of line	`^ark`	ark two	dark
$	End of line	`ark$`	noahs ark	noahs arks

Extract Phone Number

```
phone_re = r"[0-9]{3}-[0-9]{3}-[0-9]{4}"
text  = "Sam's number is 382-384-3840 and Mary's is 123-456-7890."
re.findall(phone_re, text)
```

```
['382-384-3840', '123-456-7890']
```

```
phone_re = r"([0-9]{3})-([0-9]{3})-([0-9]{4})"
text  = "Sam's number is 382-384-3840 and Mary's is 123-456-7890."
list = re.findall(phone_re, text)
```

```
[('382', '384', '3840'), ('123', '456', '7890')]
```

```
list[0][2]
```

```
'3840'
```

Normalize Date

```
messy_dates = '03/12/2018, 03.13.18, 03/14/2018, 03:15:2018'
regex = r'[/.:]'
string = re.sub(regex, '-', messy_dates)
```

```
'03-12-2018, 03-13-18, 03-14-2018, 03-15-2018'
```

```
string[12:20]
```

```
'03-13-18'
```

strip()

```
toc = '''
PLAYING PILGRIMS============3
A MERRY CHRISTMAS===========13
THE LAURENCE BOY============31
BURDENS=====================55
BEING NEIGHBORLY============76
'''
```

```
'\nPLAYING PILGRIMS============3\nA MERRY

55\nBEING NEIGHBORLY============76\n'
```

```
toc.strip()
```

```
'PLAYING PILGRIMS============3\nA MERRY
=55\nBEING NEIGHBORLY============76'
```

split()

```
lines = re.split('\n', toc.strip())
```

```
['PLAYING PILGRIMS============3',
 'A MERRY CHRISTMAS===========13',
 'THE LAURENCE BOY============31',
 'BURDENS=====================55',
 'BEING NEIGHBORLY============76']
```

```
split_re = r'=+'
[re.split(split_re, line) for line in lines]
```

```
[['PLAYING PILGRIMS', '3'],
 ['A MERRY CHRISTMAS', '13'],
 ['THE LAURENCE BOY', '31'],
 ['BURDENS', '55'],
 ['BEING NEIGHBORLY', '76']]
```

```
text = '''
"Christmas won't be Christmas without any presents," g
"It's so dreadful to be poor!" sighed Meg, looking dow
"I don't think it's fair for some girls to have plenty
"We've got Father and Mother, and each other," said Be
The four young faces on which the firelight shone brig
'''.strip()
little = pd.DataFrame({'sentences': text.split('\n')})
```

```
quote_re = r'"([^"]+)"'
spoken = little['sentences'].str.extract(quote_re)
little['dialog'] = spoken
```

	sentences	dialog
0	"Christmas won't be Christmas without any pres...	Christmas won't be Christmas without any prese...
1	"It's so dreadful to be poor!" sighed Meg, loo...	It's so dreadful to be poor!
2	"I don't think it's fair for some girls to hav...	I don't think it's fair for some girls to have...
3	"We've got Father and Mother, and each other,"...	We've got Father and Mother, and each other,
4	The four young faces on which the firelight sh...	We haven't got Father, and shall not have him ...