Answers to Exercises

Challenge 1

In Python:

```
>>> for row in data:
...     match = re.search('[^,]*(university|college|professor|dean)[^,]*',
row[-1])
...     if match: print(match.group())
```

```
 professor
 professor
 professor
 chancellor/dean
 college instructor
 professor
```

In R:

```
> matches <- regexpr('[^,]*(university|college|professor|dean)[^,]*',
data[,ncol(data)])
> regmatches(data[,ncol(data)], matches)
```

```
[1] " professor"          " professor"          " professor"          "
chancellor/dean"
[5] " college instructor" " professor"
```

Challenge 2

In Python:

```
>>> for row in data:
...     if re.search('university|college|professor|dean', row[-1]):
...         row.append(1)
...     else:
...         row.append(0)
...     print(row[-2:])
```

```
['economist, business executive, cabinet minister', 0]
['military officer', 0]
['legislator, party leader', 0]
['economist, consultant for international organizations', 0]
['lawyer, legislator, party leader, cabinet minister', 0]
['mathematician, professor, dean', 1]
['economist, professor, cabinet minister', 1]
['physician, professor, government agency official, cabinet minister, Vice
President', 1]
['military colonel', 0]
['governor, ministry official, chancellor/dean, Vice President', 1]
['legislator, mayor, party leader, cabinet minister', 0]
['legislator, head of legislature', 0]
['head of police service, college instructor, mayor', 1]
['cabinet minister', 0]
['legislator, professor', 1]
```

In R:

```
> data <- cbind(data, 0)
> matches <- grep('university|college|professor|dean', data[,ncol(data)-1])
> data[matches, ncol(data)] <- 1
> data[,(ncol(data)-1):ncol(data)]
```

```
1                                    economist, business executive, cabinet
minister 0
2                                                                  military
officer 0
3                                                        legislator, party
leader 0
4                              economist, consultant for international
organizations 0
5                                 lawyer, legislator, party leader, cabinet
minister 0
6                                                    mathematician,
professor, dean 1
7                                               economist, professor, cabinet
minister 1
8  physician, professor, government agency official, cabinet minister, Vice
President 1
9                                                                  military
colonel 0
10                        governor, ministry official, chancellor/dean, Vice
President 1
11                                 legislator, mayor, party leader, cabinet
minister 0
12                                                 legislator, head of
legislature 0
13                                 head of police service, college instructor,
mayor 1
14                                                                 cabinet
minister 0
15                                                           legislator,
professor 1
```

Challenge 3

In Sublime:

- To abbreviate years to only two digits (e.g. 1/15/2000 to 1/15/00):
    - Find: `([0-9]{1,2})/([0-9]{1,2})/[0-9]{2}([0-9]{2})`
    - Replace: `$1/$2/$3`
- To add a leading zero (e.g. 1/15/2000 to 01/15/2000):
    - Find: `([0-9])/([0-9]{1,2})/([0-9]{2,4})`
    - Replace: `0$1/$2/$3`
- To swap the month and date (e.g. 1/15/2000 to 15/1/2000):
    - Find: `([0-9]{1,2})/([0-9]{1,2})/([0-9]{2,4})`
    - Replace: `$2/$1/$3`

In Python:

```
# To abbreviate years to only two digits (e.g. 1/15/2000 to 1/15/00):
date_str = '1/15/2000'
re.sub('([0-9]{1,2})/([0-9]{1,2})/[0-9]{2}([0-9]{2})', '\\1/\\2/\\3', date_str)

# To add a leading zero (e.g. 1/15/2000 to 01/15/2000):
re.sub('([0-9])/([0-9]{1,2})/([0-9]{2,4})', '0\\1/\\2/\\3', date_str)

# To swap the month and date (e.g. 1/15/2000 to 15/1/2000):
re.sub('([0-9]{1,2})/([0-9]{1,2})/([0-9]{2,4})', '\\2/\\1/\\3', date_str)
```

In R:

```
# To abbreviate years to only two digits (e.g. 1/15/2000 to 1/15/00):
date_str <- '1/15/2000'
gsub('([0-9]{1,2})/([0-9]{1,2})/[0-9]{2}([0-9]{2})', '\\1/\\2/\\3', date_str)

# To add a leading zero (e.g. 1/15/2000 to 01/15/2000):
gsub('([0-9])/([0-9]{1,2})/([0-9]{2,4})', '0\\1/\\2/\\3', date_str)

# To swap the month and date (e.g. 1/15/2000 to 15/1/2000):
gsub('([0-9]{1,2})/([0-9]{1,2})/([0-9]{2,4})', '\\2/\\1/\\3', date_str)
```