DS 100: Principles and Techniques of Data Science **Date: July 08, 2019**

Discussion #5 Worksheet

Name:

Regular Expressions

Here's a complete list of metacharacters:

`. ^ $ * + ? { } [ ] \ | ( )`

Some reminders on what each can do (this is not exhaustive):

"^" matches the position at the beginning of string (unless used for negation "[^]")

"$" matches the position at the end of string character.

"?" match preceding literal or sub-expression 0 or 1 times. When following "+" or "*" results in non-greedy matching.

"+" match preceding literal or sub-expression *one* or more times.

"*" match preceding literal or sub-expression *zero* or more times

"." match any character except new line.

"[ ]" match any one of the characters inside, accepts a range, e.g., "[a-c]".

"( )" used to create a sub-expression

"\d" match any *digit* character. "\D" is the complement.

"\w" match any *word* character (letters, digits, underscore). "\W" is the complement.

"\s" match any *whitespace* character including tabs and newlines. \S is the complement.

"\b" match boundary between words

Some useful `re` package functions:

`re.split(pattern, string)` split the `string` at substrings that match the `pattern`. Returns a list.

`re.sub(pattern, replace, string)` apply the `pattern` to `string` replacing matching substrings with `replace`. Returns a string.

Kernel Density Estimation

1. We wish to compare the results of kernel density estimation using a gaussian kernel and a boxcar kernel. For $\alpha > 0$, which of the following statements are true? Choose all that apply.

   Gaussian Kernel:

   $$K_\alpha(x, z) = \frac{1}{\sqrt{2\pi\alpha^2}} \exp\left(-\frac{(x-z)^2}{2\alpha^2}\right)$$

   Box Car Kernel:

   $$B_\alpha(x, z) = \begin{cases} \frac{1}{\alpha} & \text{if } -\frac{\alpha}{2} \le x - z \le \frac{\alpha}{2} \\ 0 & \text{else} \end{cases}$$

   A. Decreasing α for a gaussian kernel decreases the smoothness of the KDE.

   B. The gaussian kernel is always better than the boxcar kernel for KDEs.

   C. Because the gaussian kernel is smooth, we can safely use large α values for kernel density estimation without worrying about the actual distribution of data.

   D. The area under the box car kernel is 1, regardless of the value of α.

   E. None of the above.

Regular Expressions

2. Which strings contain a match for the following regular expression, `"1+1$"`? The character `"␣"` represents a single space.

   ○ `What␣is␣1+1` ○ `Make␣a␣wish␣at␣11:11` ○ `111␣Ways␣to␣Succeed`

3. Given the text:

   ```
   "<record>␣Samuel␣Lau␣<samlau95@gmail.com>␣Faculty␣</record>"
   "<record>␣Manana␣Hakobyan␣<manana.hakobyan@berkeley.edu>␣TA␣</record>"
   ```

   Which of the following matches exactly to the email addresses (including angle brackets)?

   ○ `<.*@.*>` ○ `<[^>]*@[^>]*>` ○ `<.*@\w+\..*>`

4. For each pattern specify the starting and ending position of the first match in the string. The index starts at zero and we are using closed intervals (both endpoints are included).

	abcdefg	abcs!	ab␣abc	abc,␣123
abc*	$[0, 2]$			
[^\s]+				
ab.*c				
[a-z1,9]+				

5. Write a regular expression that matches strings (including the empty string) that only contain lowercase letters and numbers.

6. Write a regular expression that matches strings that contain exactly 5 vowels.

7. Given that `address` is a string, use `re.sub` to replace all vowels with a lowercase letter “o”. For example `"123␣Orange␣Street"` would be changed to `"123␣orongo␣Stroot"`.

8. Given `dates = "October␣10,␣November␣11,␣December␣12,␣January␣1"`, use `re.findall` to extract all the numbers in the string. The result should look like **`["10", "11", "12", "1"]`**.

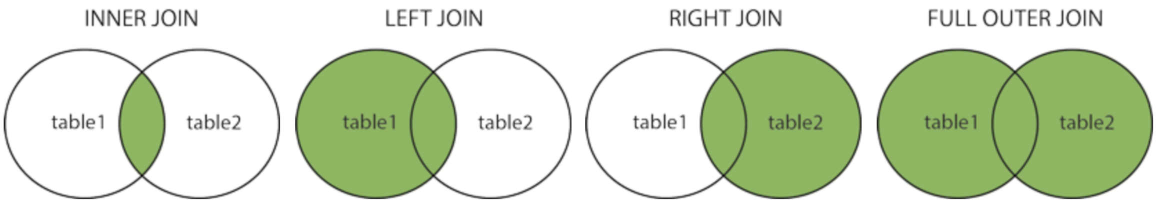
9. Given the following text in a variable `log`:

```
169.237.46.168 - - [26/Jan/2014:10:47:58 -0800]
"GET␣/stat141/Winter04/␣HTTP/1.1" 200 2585
"http://anson.ucdavis.edu/courses/"
```

Fill in the regular expression in the variable `pattern` below so that after it executes, day is 26, month is Jan, and year is 2014.

```
pattern = ...
matches = re.findall(pattern, log)
day, month, year = matches[0]
```

SQL



Note: You do not always have to use the JOIN keyword to join sql tables. The following are equivalent:

```
SELECT column1, column2
FROM table1, table2
WHERE table1.id = table2.id;

SELECT column1, column2
FROM table1 JOIN table2
ON table1.id = table2.id;
```

10. Describe which records are returned from each type of join.

11. Consider the following real estate schema:

```
Homes(home_id int, city text, bedrooms int, bathrooms int,
area int)
Transactions(home_id int, buyer_id int, seller_id int,
transaction_date date, sale_price int)
Buyers(buyer_id int, name text)
Sellers(seller_id int, name text)
```

Fill in the blanks in the SQL query to find the id and selling price for each home in Berkeley. If the home has not ben sold yet, **the price should be NULL**.

```
SELECT ____________________________________
FROM _________________________________________
____________________ JOIN ____________________
ON ______________________________
WHERE ______________________________;
```