Lecture 6

Recitation – 06/04/2016

Q1 Combinations & Probabilities

(a) Write a function *couples_dictionary* that receives an integer, *n*, and returns a dictionary with keys and values as follows:

- The keys are the integers between 2 and $2n$ (including both)
- The values are lists, containing all the couples of integers between 1 and n, whose sum equals to the key (treat the couples [x,y] and [y,x] as different couples).

Example:

```
>>> couples_dictionary(3)
{2: [[1, 1]], 3: [[1, 2], [2, 1]], 4: [[1, 3], [2, 2], [3, 1]], 5: [[2, 3], [3, 2]], 6: [[3, 3]]}
```

(b) Write a function *calc_key_proportion* that receives a dictionary in the format of (a), and returns a new dictionary with the same keys, but instead of the lists as values, the values would be the length of the list, divided by the sum of the lengths of all lists (the proportion of couples that their sum equals to the key).

Example:

```
>>> cd = couples_dictionary(3)   # the sum of the lengths of all lists is 9
>>> calc_key_proportion(cd)
{2: 0.1111111111111111, 3: 0.2222222222222222, 4: 0.3333333333333333, 5: 0.2222222222222222, 6:
0.1111111111111111}
```

(c) Write a function, *dice_prob*, that receives two integers, *n* and *val*, and returns the answer to the following question: what is the chance of getting val, when tossing two n-dice.

Use the functions (a) and (b) in order to calculate the answer.

Q2 The human disease ontology (http://disease-ontology.org/)

Biological ontologies are represented in an OBO file. In this format, each ontology term has its own entry. Each ontology entry starts with a line "[Term]". The lines after this line list the ontology-term information. The ontology entry ends with an empty line. The row is_a represents the terms that are the ancestors of the current Term. A DO may have one ancestor, multiple or none at all. For example:

```
[Term]
id: DOID:2649
name: chondroblastoma
synonym: "chondroblastoma" EXACT [CSP2005:2019-1220]
synonym: "Chondroblastoma (morphologic abnormality)" EXACT [SNOMEDCT_2005_07_31:9001003]
synonym: "Chondroblastoma NOS (morphologic abnormality)" EXACT [SNOMEDCT_2005_07_31:189887007]
synonym: "Chondroblastoma of bone" EXACT [SNOMEDCT_2005_07_31:134337007]
xref: MSH:D002804
xref: NCI:C2945
xref: SNOMEDCT_2010_1_31:134337007
xref: SNOMEDCT_2010_1_31:189887007
xref: SNOMEDCT_2010_1_31:9001003
xref: UMLS_CUI:C0008441
is_a: DOID:201 ! connective tissue cancer
```

Has 1 ancestor

```
[Term]
id: DOID:265
name: spleen angiosarcoma
def: "An angiosarcoma and hemangioma of intra-abdominal structure and malignant soft tissue neoplasm of the spleen that resu
synonym: "angiosarcoma of spleen (disorder)" EXACT [SNOMEDCT_2005_07_31:187821001]
synonym: "Splenic hemangiosarcoma" EXACT [NCI2004_11_17:C4564]
xref: NCI:C4564
xref: SNOMEDCT_2010_1_31:187821001
xref: UMLS_CUI:C0346424
is_a: DOID:0001816 ! angiosarcoma
is_a: DOID:254 ! hemangioma of intra-abdominal structure
is_a: DOID:672 ! spleen cancer
```

Has multiple ancestors

```
[Term]
id: DOID:2651
name: intraductal papillomatosis
synonym: "Intraductal papillomatosis" EXACT [NCI2004_11_17:C7363]
synonym: "Intraductal papillomatosis (morphologic abnormality)" EXACT [SNOMEDCT_2005_07_31:32296002]
synonym: "Intraductal papillomatosis NOS (morphologic abnormality)" EXACT [SNOMEDCT_2005_07_31:189710005]
xref: NCI:C7363
xref: SNOMEDCT_2010_1_31:189710005
xref: SNOMEDCT_2010_1_31:32296002
xref: UMLS_CUI:C0334377
is_obsolete: true
```

Has no ancestors

In the class files you can find the file HumanDO.obo that represents the human disease ontology.

a. Write a function *parse_obo* that receives an OBO file and returns a dictionary for which the keys are the DO terms IDs and the values are lists of the ancestors DO terms ids. For example:

```
>>> od_ancestors_dict = parse_obo("HumanDO.obo")
>>> od_ancestors_dict["DOID:2649"]
['DOID:201']
>>> od_ancestors_dict["DOID:265"]
['DOID:0001816', 'DOID:254', 'DOID:672']
>>> od_ancestors_dict["DOID:2651"]
[]
```

Note(!) In the end of the file there are a few definitions that start with [Typedef]. The lines in these sections also begin with "id: ". Beware!

b. Write a function *is_ancestor* that receives as input two DO IDs: *father*, *son*, and an obo file (the file path as a string), and returns **True** if *father* is the direct ancestor of *son*, and **False** otherwise.

Q3 CSV merge and analyze

In the folder Simulation_Raw_Data there are csv files with data collected from simulation runs.

In the simulation, a population is modeled and the data collected is:

a. The time (in generations) a mutant appears for the first time.

b. The time (in generations) the simulation ended (when the mutant fixated or got extinct).

c. Indicator of fixation:

   i. 1 if the mutant fixated in the population

   ii. 0 if the mutant got extinct.

Each csv holds the data from one simulation run, and has 2 lines:

1. The first row represents the parameters' values as a concatenated list of

   (**parameter symbol**, parameter value)

   for example: ***n,5,k,2000,m,1.00E-06,r,0.1***

2. The second line contains the results (a,b,c) as described above.

Example of a simulation csv file:

	A	B	C	D	E	F	G	H	I
1	n	5	k	2000	m	1.00E-06	r	0.1	
2	6508	6517	0						

(a) Write a function, *csv_merge*, that receives the folder's path as a string, and creates a single csv file that holds the entire data and summarize it. The new csv should look like:

	A	B	C	D	E	F	G
1	n	k	m	r	First Appe	Finish	ext/fix
2	10	1000	1.00E-05	0.1	7799	7806	0
3	5	2000	1.00E-05	0.01	5196	5197	0
4	5	2000	1.00E-06	0.1	6508	6517	0
5	5	2000	1.00E-06	0.01	9016	9024	0
6	2	5000	1.00E-06	0.1	6043	6047	0

And at the bottom:

200	10	1000	1.00E-06	0.1	7659	8124	1
201	2	5000	1.00E-05	0.1	922	928	0
202							
203	Mean First Appearance	Mean Finish	Fixation Probability				
204	4869.4	4952.37	0.325				

To get a list of all file names inside a folder use the function: os.listdir(*path*)

(b) Write a function, *find_means*, that receives the path of such a summary file (like the one you create in (a)) and values of n, k, m & r, and returns the mean of the First Appearance and the mean survival rate (mean of ext/fix), for the simulations with same n,k,m,r values.

Example:

```
>>> find_means('summary.csv',10,1000,10**-6,0.01)
(5161.117647058823, 0.29411764705882354)
```