1. Data downloaded from Keck site as .tbl.txt
2. Data converted into excel spreadsheet
3. Python Script Launched

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Convert data.xlsx to lists for each column using modified version of

import csv

with open('so.csv') as f:

bl = [[],[],[],[],[]]

reader = csv.reader(f)

for row in reader:

for col in range(5):

bl[col].append(row[col])

Name each list the title of each column

(http://s tackoverflow.com/questions/11263014/converting-columns-in-a-file-into-separate-lists-using-python)

1. Ask user what they want to check

Sample: If input=I, while….

* + 1. Numbers of Each
       1. Number of different PI’s
       2. Number of unique projects
       3. Number of unique nights for a given project
    2. Lists of Each
       1. List of Projects Under Specific PI
       2. List of unique PI’s
       3. List of unique projects
       4. List of unique Target Names under one project
    3. Optional extra to do later (not a priority)
       1. Total Exposure for specific project

When Python 2.7 comes out you can use its [collections.Counter class](http://docs.python.org/dev/library/collections.html#collections.Counter)

otherwise see [counter receipe](http://code.activestate.com/recipes/576611/)

Under Python 2.7a3

rom collections import Counter

input = ['a', 'a', 'b', 'b', 'b']

c = Counter( input )

print( c.items() )

output is

[('a', 2), ('b', 3)]

(<http://stackoverflow.com/questions/2392929/how-to-get-unique-values-with-respective-occurance-count-from-a-list-in-python>)

If asked, what are all the projects under a specific PI:

Search PI’s name in list

Get position in list

Look up that position number in transposed list

Get number of name of project in transposed list

Make list of all the projects

Get set of that list, to get unique names

Print unique names

If asked for number of projects under a specific PI:

Get length of list

Filtering a collection

That is, finding all elements in a sequence that meet a certain condition. You can use list comprehension or generator expressions for that:

matches = [x for x in lst if fulfills\_some\_condition(x)]

matches = (x for x in lst if x > 6)

The latter will return a *generator* which you can imagine as a sort of lazy list that will only be built as soon as you iterate through it. By the way, the first one is exactly equivalent to

matches = filter(fulfills\_some\_condition, lst)

in Python 2. Here you can see higher-order functions at work. In Python 3, filter doesn't return a list, but a generator-like object.

Finding the first occurrence

If you only want the first thing that matches a condition (but you don't know what it is yet), it's fine to use a for loop (possibly using the else clause as well, which is not really well-known). You can also use

next(x for x in lst if ...)

which will return the first match or raise a StopIteration if none is found. Alternatively, you can use

next((x for x in lst if ...), [default value])

(<http://stackoverflow.com/questions/9542738/python-find-in-list>)