




Assignment 2

Assignment 2



Assignment 2

Attached Files:  [CrimeDataSet.csv](#) (894.105 KB)
 [CrimeDataSetDirty.csv](#) (894.349 KB)
 [CrimeDataSmall.csv](#) (1.963 KB)
 [CrimeDataSmallDirty.csv](#) (1.995 KB)
 [Assignment2.py](#) (1.563 KB)

This assignment is worth 20 marks, and will count as **20% of your final mark** in this subject.

There are FIVE (5) questions in this assignment. The fifth question will require you to call the functions you wrote in the first four questions

Things to look out for in solving the questions are:

- never be afraid to create extra variables, e.g. to break up the code into conceptual sub-parts, improve readability, or avoid redundancy in your code
- we also encourage you to write helper functions to simplify your code – you can write as many functions as you like, as long as one of them is the function you are asked to write
- commenting of code is one thing that you will be marked on; get some practice writing comments in your code, focusing on:
 1. describing key variables when they are first defined (but not things like index variables in **for** loops)
 2. describing what "chunks" of code do (i.e. not every line, but chunks of code that perform a particular operation, such as **#find the maximum value in the list** or **#count the number of vowels**)

The Australian crime statistics database holds crime statistical data that is freely available on the Australian government website: data.gov.au/dataset. The data indicates trends in crime covering the whole of Australia, which is separated between counties and Local Government Authority areas (LGA), over a number of years. The information held in these databases highlight the number of crimes committed from Trespass to Homicide, in a number of geographical locations.

Congratulations! You have been appointed by the MUC Dept. in the Australian Government to help them ascertain various information within the crime dataset. The dataset has been vandalised by high tech criminals, and will need your skills to help clean it up, prior to providing basic analysis. As part of your task, you are asked to write four (4) functions that perform specific tasks, plus a "main" function that will utilise all the other functions.

The Crime Statistics data is given to you in one or more **comma-separated values (CSV)** files. You will find this in the Assignment 2 folder on LMS.

CSV is a simple file format which is widely used for storing **tabular data** (data that consists of columns and rows). In CSV, columns are separated by commas, and rows are separated by newlines (so every line of the text file corresponds to a row of the data).

Usually, the first row of the file is a header row which gives names for the columns.

The Crime Statistics data contains the following columns:

ID

An integer unique ID assigned to each row of data.

Statistical Division or Subdivision

The broad area the crimes were committed.

LGA

The Local Governance Area that managed the crime area.

Offence category

A title of the crime category.

Subcategory

A breakdown of the crime within each category area.

Year statistics (from 2002 through to 2012)

Holding a tally of each crime took place in that year.

Provided is a sample of the CVS data provided to you by the MUC (in fact we have provided 4 data samples, two large and two small, one of each are contaminated (dirty), and the others are clean)

In order to clean up and analyse the data, we need a way to take data from a CSV file and put it into a Python data structure. Fortunately, Python has a built-in CSV library which can do most of the work for us.

You won't have to use the **csv** library directly, though. We will provide you with a helper function called **read_data** which uses the **csv** library to read the data and turn it into a **dictionary of dictionaries**. For example, suppose the data above was stored in a file called **CrimeDataSet.csv**. To work with this data in Python, we would call

```
read_data("CrimeDataSet.csv")
```

which would return the following Python dictionary:

```
{ '1' : {'Division': 'Inner Sydney', 'LGA': 'Botany Bay', 'Offence': 'Homicide', 'Subcategory': 'Murder (a)', '2002': '1', '2003': '0', '2004': 'zero', '2005': '1', '2006': '2', '2007': '1', '2008': '0', '2009': '1', '2010': '0', '2011': '0', '2012': '1'}}
```

Note

Notice that all of the values in the nested dictionaries are strings, even the numeric values. If you want to use the values in numerical calculations, you will have to typecast them yourself.

Nested dictionaries can be confusing. Here are some simple examples of how to access data in a nested dictionary:

```
# save the data in a variable
```

```
data = { '1' : {'Division': 'Inner Sydney', 'LGA': 'Botany Bay', 'Offence': 'Homicide', 'Subcategory': 'Murder (a)', '2002': '1', '2003': '0', '2004': '0', '2005': '1', '2006': '2', '2007': '1', '2008': '0', '2009': '1', '2010': '0', '2011': '0', '2012': '1'}}
```

```
# Where is the '1' ID's Division
```

```
print(data["1"]["Division"])
```

```
# What is the second ID's subcategory
```

```
print(data["2"]["Subcategory"])
```

```
# What is the summation of each year of ID '1'
```

```
s = 0
```

```
for i in range(2002, 2012+1):
    s += int(data['1'][str(i)])
```

You have been provided with large CSV files containing Crime data within Australia. Unfortunately, the data is "noisy": some people have attacked the data form, or intentionally entered incorrect data, to subvert the clear understanding of crime. Your first task as a programmer-analyst is to clean up the noisy data for later analysis.

There are a few particular errors in this data:

- Criminals have altered the data to include zero's instead of 0, in the years, or entered null (or even negative!) as a means to damage the dataset. All year values should be greater than or equal to 0.

Question 1 (Clean data)

Write a function **clean_data(data)** which takes one argument, a dictionary of data in the format returned by **read_data**. This data has been read directly from a CSV file, and is noisy! Your function should construct and return a new data dictionary which is identical to the input dictionary, except that invalid data values have been replaced with **0**. **You should not modify the argument dictionary, data**. The cleaning process should keep a count of all the data samples it cleans, and **return** the summated value of how many it cleaned.

For example, let's look at the data contained in **CrimeDataSetDirty.csv**:

```
{'1' : {'Division': 'Inner Sydney', 'LGA': 'Botany Bay', 'Offence':
'Homicide', 'Subcategory': 'Murder (a)', '2002': '1', '2003': '0',
'2004': 'zero', '2005': '1', '2006': '2', '2007': '1', '2008': '0',
'2009': '1', '2010': '0', '2011': '0', '2012': '1'}}
```

Clearly some of the values are invalid! Let's call **clean_data** on the data, and look at the result:

```
{'1' : {'Division': 'Inner Sydney', 'LGA': 'Botany Bay', 'Offence':
'Homicide', 'Subcategory': 'Murder (a)', '2002': '1', '2003': '0',
'2004': '0', '2005': '1', '2006': '2', '2007': '1', '2008': '0', '2009':
'1', '2010': '0', '2011': '0', '2012': '1'}}
```

Notice the **0** values in the nested 2004 dictionary of the cleaned data, was previously 'zero'.

You can assume the following:

- the input data dictionary should not contain **zero** or **null** values;
- all year stats (once cleaned) are strings that can be cast to **ints**;

Question 2 (Worst year)

Write a function called **countCrimes(data, key)** which takes a dictionary containing crime data and a dictionary key; for all year statistics data. The function summates all the values within that key and returns the finally value.

For example, all crimes for key '2012' should return an integer representing the total sum of crimes though all ID rows for that year.

Using this value, you must calculate the worst year for crime, (i.e. the year with the maximum total crimes) and print out the year and crime number in that year. You may assume the crime data in **data** is "clean" after invalid values have been replaced by **0**. If a nested dictionary contains a **None** value for the **year** key, you should ignore it in your calculation. You may also assume that all values are positive integers, and all cleaned data values have been repaired. A **clean_data_set.csv** file containing clean data is also supplied to allow you to test this method, in case you were unable to clean the data independently.

Question 3 (Worst area)

Your employers are interested in the distribution of crime throughout the different Statistical Subdivision areas. One way to establish this is to divide each **Subdivision** into unique bins or

dictionary keys; where a key holds a summation of all the crimes for all the years within that subdivision area.

Write a function called **worstCrime(data)**, which summates the values of each Subdivision of each year and returns a new Dictionary, where the key is the Subdivision name and the value is the summated total of all crimes within that area over all years. Display the number of Subdivisions found, and present the area with the highest overall crime values as the 'Worst area'.

Question 4 (Most active criminal activity)

The MUC are interested in learning which crime is 'persued' the most throughout the whole dataset. By acquiring this information, they will be able to focus on reinforcing security levels, targeting that type of crime more robustly. You have been tasked with providing this information.

Write a function called **mostActive(data)**, which returns a dictionary of the different crime types, holding a tally of how often those crimes were committed overall. That is, each key within the dictionary will be the name of a crime; such as Homicide, or Robbery, etc., and the values therein, will be the tally of crimes for that particular crime throughout all years. Finally, from the returned dictionary, find the worst crime type and present it as the 'Most active Crime overall', and include the summated value.

Question 5 (Tying it all together)

A prestigious Victorian university has asked the MUC to produce a report on the final status and crime situation within the dataset. They have asked you to help them generate data for this report.

Write a function called **main(datafile)** which takes a filename as an argument, which reads the crime data contained in that file, cleans the data, and uses the data to print out some facts about crime in Australia. You should assume that the data in **datafile** is noisy. Your function should calculate and print out the following facts:

- The total number of rows in the data file
- The total number of Subdivisions examined in the data
- The total number of Offence Categories
- The worst area for crime
- The most active type of crime.

Note

You will probably find it useful to call **read_data**, **clean_data**, etc. in your **main** function.

Present your analysis in a single formatted text output, with the statistical data values embedded. Here is an example of what the output might look like; XXXXX has been used to represent your analytical data, where **Your name** is written, please add your name.

```
'On behalf of the MUC (Made Up Company), I, Your Name, have analysed  
XXXXX units of the crime statistics data. This data covered XXXXX  
Subdivisions and found XXXXX types of crimes. I conclude that the worst  
area for crime is XXXXX, and that the most active category of crime is  
XXXXX.'
```

Save your Python file as **Assignment2.py**. To submit your work, please upload your Python Assignment2.py file, containing your complete source code (with all 5 functions) through the LMS turnitin platform.



Assignment 2

>> [View/Complete](#)