Project Stage 2: Data Analysis

- INFO1903
- Rahul Vemulapally
- SID - 440317463

PART 1

Aim

The aim of this stage is to rank the LGAs (Local Government Area) based on the reported crimes in NSW.

Data Sets

Stage one cleaned the reported crimes data that was provided by NSW Bureau of Crime Statistics and Research (BOCSAR). Data Source: www.bocsar.nsw.gov.au Or source_files/RCI_offencebymonth.xlsm

- The original data was from Jan,1995 - Dec,2016; for a total of 264 periods
- Monthly counts were added together for each year, to give 22 periods of data (1995 - 2016).

- Original data had 62 different categories/sub-categories, these were grouped into 11 main categories.

- The 11 main Categories are:

Main Categories
Homicide
Assault
Sexual offences
Other offences against the person
Robbery
Property Offences
Drug Possession
Drug Dealing
Vice
Weapons offences
Other offences

To see how the 62 categories were grouped into these 11 main categories, please see offence_categories.xlsx.

The cleaned data is in the file NSWcrimes.csv , it lists the reported crimes for 62 different categories/sub-categories for the years 1995-2016. The category and Subcategory are in two different fields, when processing the file NSWcrimes.csv, these are concatenated together to get the 62 categories.

Two more data sets were used in this analysis:

- **Population Counts for each LGA:**

    - This data was obtained from ABS 2011 census by building the following table:
        - 2011 Census of Population and Housing - TableBuilder Basic
        - 2011 Census - Persons and Relationships
        - Table: LGA by GCCSA
            - LGA: Local Government Areas (2011 Boundaries) > New South Wales > LGA
            - GCCSA: Greater Capital City Statistical Areas (UR) > New South Wales > GCCSA
            - This table lists the "Persons' Place of Usual Residence"

- This data is stored in source_files/LGApop2011.xlsx

- The total population counts for each LGA were taken and saved to [2011LGA_pop.csv]

- **Rents dataset:**

    - Holds the rents for 1,2,3,4+ bedroom dwellings for each LGA in GMR (Greater Metropolitan Regions, i.e., most of Greater Sydney and some regional cities), [Source: www.housing.nsw.gov.au].
    - Cleaned and saved the data in "lga_ad" tab from the original file to [rents_gmr.csv].

- [lga.txt] : holds the names of all 152 LGAs plus (Unincorporated Far West, Lord Howe Island, Prisons)

Two methods were used in the analysis.

Before we proceed further it is important to note that Sydney LGA has the highest aggregate crimes reported, this is characteristic of any area that has a high transient population (visitors). To make any meaningful comparisons we will have to add this transient population to the resident population but deciding on an appropriate number for the transient population is very subjective and choosing a large number will significantly improve Sydney's ranking!

Method-1

Done using absolute counts and also counts adjusted for population in each LGA:

- Used 2011 Census population counts (for each LGA) throughout the analysis, it would have been appropriate to use Census data from different years to improve accuracy but due to time constraints I will be using 2011 population counts for all 22 years of data.
- Adjusted counts are for every 10000 people in that LGA and calculated as:

$$\left[\frac{\text{LGA's aggregate for each Year}}{\text{2011 population for the corresponding LGA}} \right] * 10000$$

- **Psudocode**

- For each LGA:

    - For each Year:
        - Add all the crimes reported (i.e., add all categories' counts for each year)
        - Rank the LGA's from lowest to highest counts (Table 1 in the below illustration)

- This should give 22 ranks (if all 22 Years selected) for each LGA
- Take the mean/median of these ranks to get the overall rank for each LGA (Table 2 in the below illustration)

- An illustration of ranking population adjusted counts.

	Ashfield	Auburn	Bankstown	Blacktown	Blue Mountains	Botany Bay	Burwood	Camden
1995	863	619	754	764	649	1000	1063	290
1996	922	765	898	845	626	1145	1169	360
1997	1038	832	988	970	796	1117	1259	401
1998	1040	884	1067	984	858	1218	1155	418
1999	1054	966	1036	1016	764	1122	989	409
2000	1117	1084	1106	1126	801	1245	1176	515
2001	1057	1065	1097	1136	829	1264	1412	617
....	...	...	...	...	...	...	.	..
....	...	...	...	...	...	...	...	...
2013	833	1155	980	1198	659	874	1112	531
2014	756	1176	1015	1267	779	913	1215	550
2015	808	1201	993	1584	833	948	1141	661
2016	739	969	1015	1523	636	875	1025	711

1

Rank across each row from lowest to highest counts for all LGAs.

Which gives the below table of ranks.

	Ashfield	Auburn	Bankstown	Blacktown	Blue Mountains	Botany Bay	Burwood	Camden
1995	107	55	89	94	65	128	133	4
1996	102	76	98	90	41	128	132	7
1997	106	77	99	96	73	122	133	7
1998	98	76	100	85	68	119	113	6
1999	99	83	95	90	41	108	86	5
....	...	...	...	...	...	..	.	...
....	...	...	...	...	...	..	.	...
2013	67	117	91	119	42	77	109	23
2014	53	117	96	123	58	84	118	26
2015	62	113	92	137	68	84	108	47
2016	55	89	93	139	35	74	96	50

2

22 rows × 153 columns

Then take median/mean across each column/LGA to get the overall rank

This method treats all crimes with equal weightage, which is erroneous!

Method-2 corrects this by ranking LGA's separately for each category and taking the mean rank over all the categories to get the overall rank.

Method-2

- Only used the population adjusted counts.

- Psudocode
- For each Category:
    - For each Year:
        - Rank the LGA's from lowest to highest counts
    - This should give 22 ranks (if all 22 Years selected) for each LGA (Table A in the below illustration)
    - Take the mean of these ranks to get the overall rank for each LGA for that Category (e.g: In the below Final Table, Assault row ranks are the means from Table A)
- This should give 11 ranks (If all 11 categories selected) for each LGA
- Take the mean of these 11 ranks to get an overall rank for each LGA.
- An illustration of calculating the ranks for each Category under Method-2

A

Below are the ranks for the Category "Assault" for each Year.

	Albury	Armidale Dumaresq	Ashfield	Auburn	Ballina	Balranald	Bankstown
1995	125	87	59	68	42	105	84
1996	118	54	40	68	69	92	79
1997	119	85	59	82	63	135	77
....	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...
2014	127	131	28	88	72	123	75
2015	121	126	51	82	48	70	78
2016	116	118	31	77	51	76	81

22 rows × 153 columns

This is how the final table rankings are calculated for each category.

Taking the mean across each column gives the mean rank for the Category "Assault" for each LGA

Final Table:

	Albury	Armidale Dumaresq	Ashfield	Auburn	Ballina	Balranald	Bankstown
Homicide	133	131	134	131	134	135	136
Assault	114	105	42	65	73	124	62
Sexual offences	110	109	51	43	78	111	42
Other offences against the person	100	80	54	72	58	120	89
Robbery	101	102	141	147	83	45	140
Property offences	131	121	86	89	70	100	91
Drug Possession	97	94	69	108	61	129	66
Drug Dealing	104	92	56	78	73	131	62
Vice	122	90	40	44	78	118	24
Weapons offences	108	64	43	64	40	130	61
Other offences	101	90	72	107	47	130	70

11 rows × 153 columns

Exploring the Data

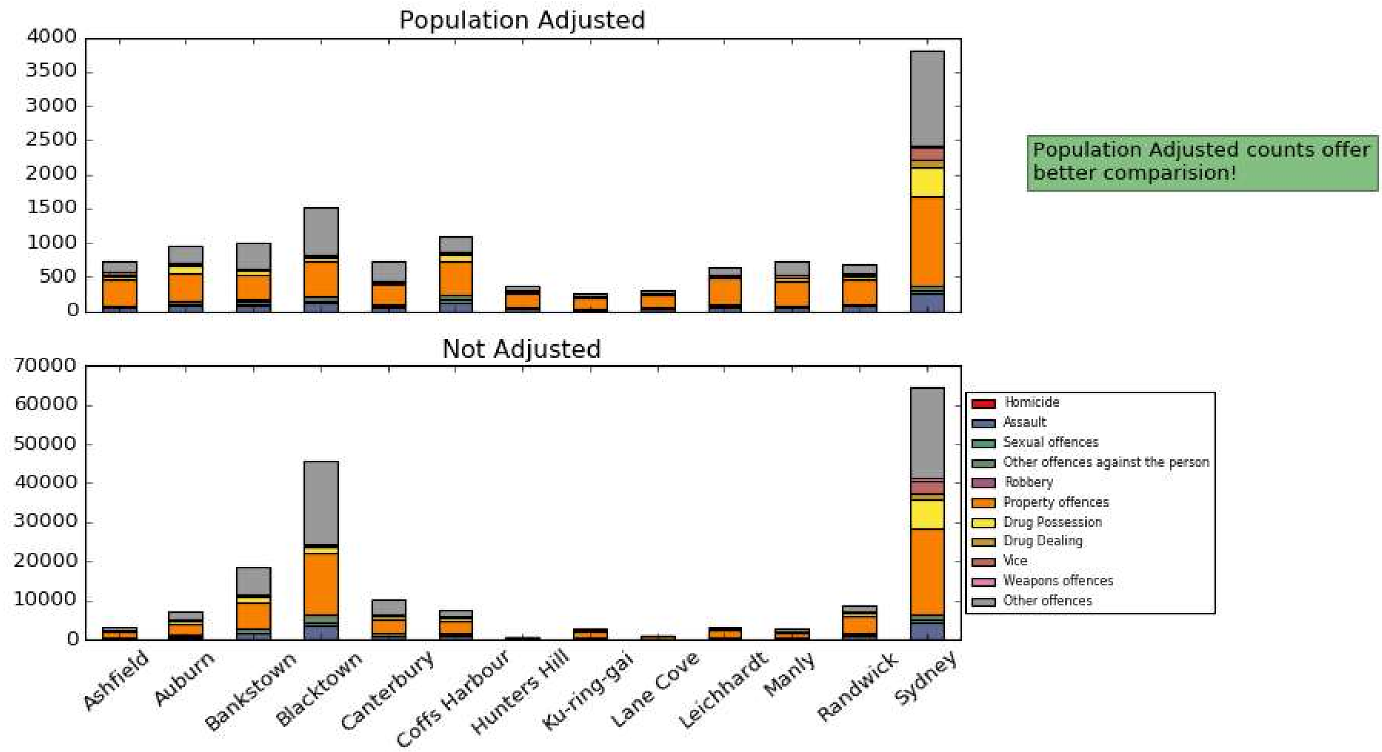
Here is the data for the year 2016, for some of the LGAs:

Categories	Ashfield	Auburn	Bankstown	Blacktown	Canterbury	Coffs Harbour	Hunters Hill	Ku-ring-gai	Lane Cove	Leichhardt	Manly	Randwick	Sydney
Homicide	0	0	8	6	5	0	0	0	0	0	0	2	3
Assault	213	624	1578	3710	866	920	38	152	93	327	242	906	4532
Sexual offences	56	88	188	556	126	203	11	85	15	38	44	160	548
Other offences against the person	101	345	1099	1999	466	544	28	96	53	170	46	342	1176
Robbery	7	39	65	251	65	21	3	6	1	16	4	48	264
Property offences	1497	2914	6748	15664	3856	3298	266	1859	584	2033	1437	4438	21883
Drug Possession	271	952	1310	1375	526	592	35	96	46	130	137	825	7316
Drug Dealing	62	89	214	266	99	151	4	42	4	31	28	154	1760
Vice	134	123	66	231	72	113	6	4	2	34	180	288	3012
Weapons offences	62	118	307	510	138	174	5	34	2	45	23	94	679
Other offences	643	1852	6928	21286	4017	1485	102	451	152	525	783	1608	23527

Here is the population adjusted data for the year 2016, for the same LGAs:

Categories	Ashfield	Auburn	Bankstown	Blacktown	Canterbury	Coffs Harbour	Hunters Hill	Ku-ring-gai	Lane Cove	Leichhardt	Manly	Randwick	Sydney
Homicide	0	0	0	0	0	0	0	0	0	0	0	0	0
Assault	51	84	86	123	63	134	28	13	29	62	60	70	267
Sexual offences	13	11	10	18	9	29	8	7	4	7	11	12	32
Other offences against the person	24	46	60	66	33	79	21	8	16	32	11	26	69
Robbery	1	5	3	8	4	3	2	0	0	3	1	3	15
Property offences	363	395	370	520	280	482	201	170	185	389	361	344	1290
Drug Possession	65	129	71	45	38	86	26	8	14	24	34	63	431
Drug Dealing	15	12	11	8	7	22	3	3	1	5	7	11	103
Vice	32	16	3	7	5	16	4	0	0	6	45	22	177
Weapons offences	15	16	16	16	10	25	3	3	0	8	5	7	40
Other offences	156	251	379	706	292	217	77	41	48	100	196	124	1387

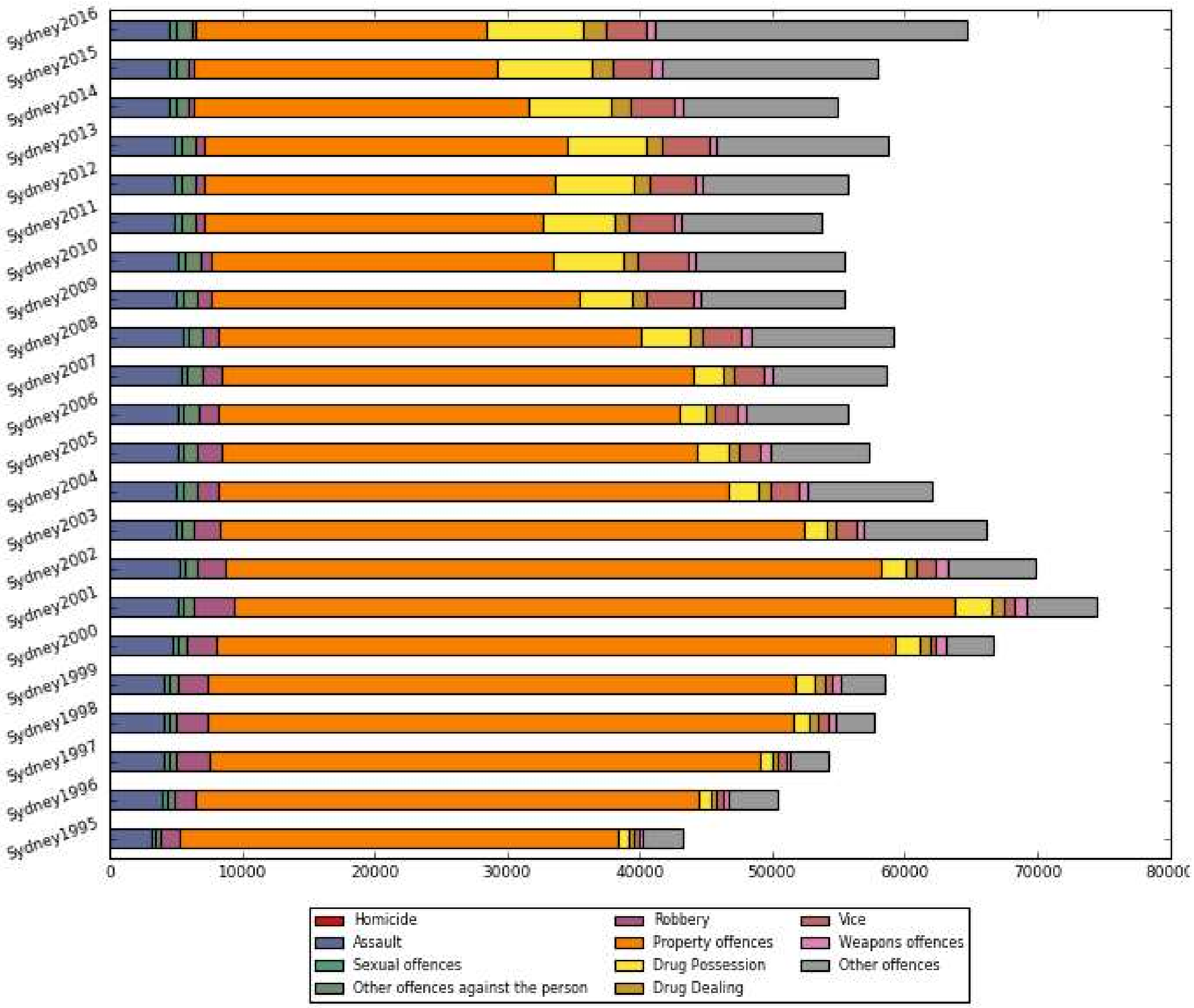
Here's a visual comparison of the above two tables using stacked bar charts



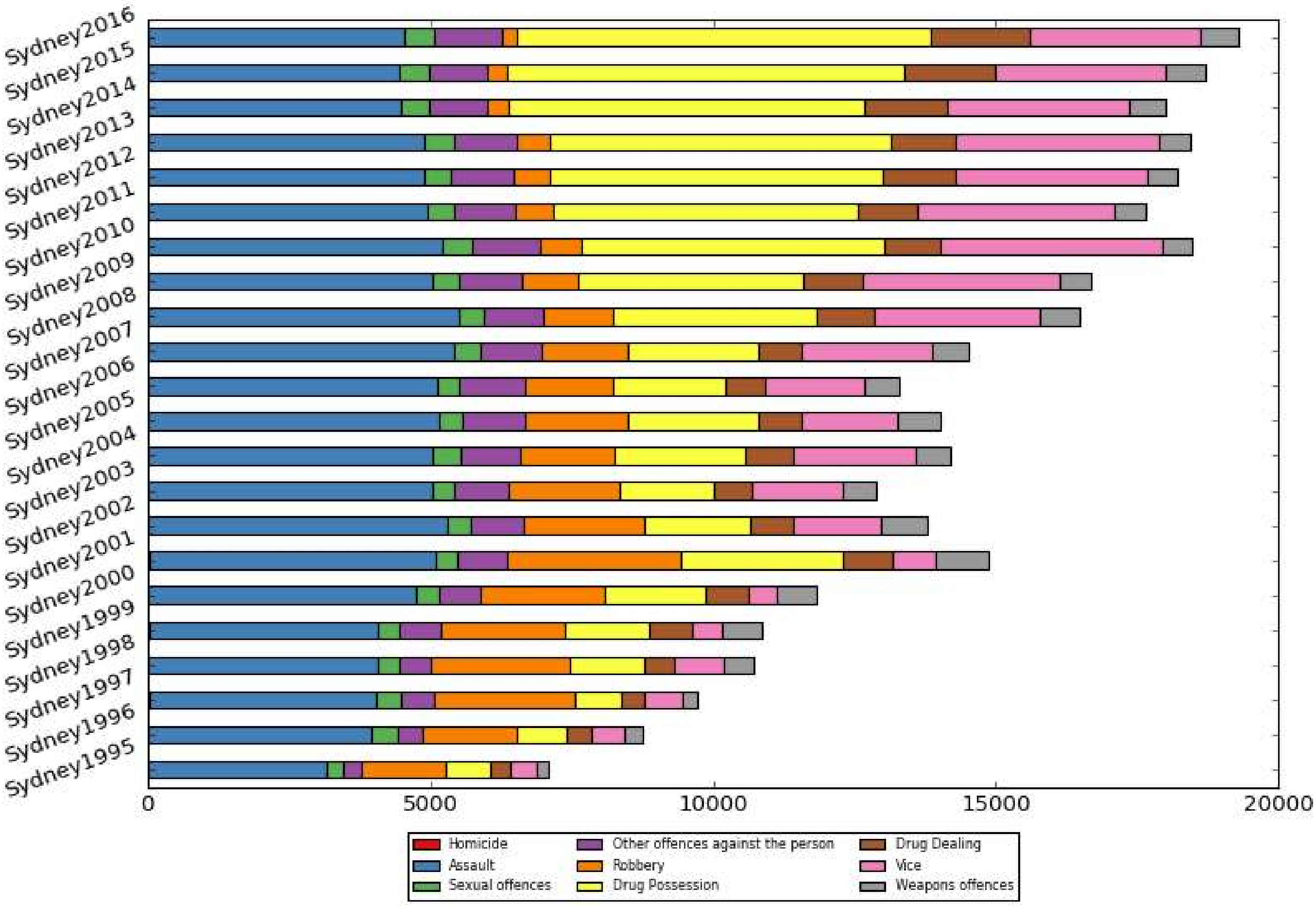
Note: As mentioned earlier, comparing Sydney with other LGAs is not appropriate.

Exploring stats for a single LGA:

- Here is a stacked bar plot of Sydney showing all categories.



- Here is the stacked bar plot of Sydney without "Property Offences" and "Other offences"



It is very clear that "Drug Possession" crimes have increased quite a bit over the last few years.

Analysis

Methond-1 analysis

Explanation of the different ranks

- sum_median_rank: Data is not population adjusted, and ranked by using the median rank
- sum_mean_rank: Data is not population adjusted, and ranked by using the mean rank
- adj_median_rank: Data is population adjusted, and ranked by using the median rank
- adj_mean_rank: Data is population adjusted, and ranked by using the mean rank
- y2011POP: 2011 Population for each LGA

All 153 LGAs

- The below table shows the Top-10 LGA's
- Sorted by 'adj_mean_rank'

TOP-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP
Ku-ring-gai	100	101	2	4	109300
The Hills Shire	124	125	4	4	169873
Unincorporated Far West	1	1	4	6	1168
Palerang	36	36	6	6	14351
Dungog	22	23	8	11	8316
Cabonne	36	36	11	12	12823
Wollondilly	78	78	14	14	43261
Hornsby	133	132	14	15	156849
Upper Lachlan Shire	24	24	18	17	7192
Gloucester	14	14	15	18	4879

- And the Bottom-10

Bottom-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP
Bourke	70	71	153	153	2867
Brewarrina	43	45	152	152	1766
Central Darling	42	44	151	151	1992
Sydney	153	153	150	150	169507
Walgett	78	76	149	149	6453
Moree Plains	95	97	148	148	13428
Coonamble	52	52	147	147	4031
Narrandera	54	55	143	140	5900
Dubbo	122	122	142	140	38808
Wellington	66	65	140	137	8494

Greater Sydney - 43 LGAs

- Sorted by adj_mean_rank
- one_bed, two_bed, three_bed, and four_plus_bed are the median rents in Dollars from the last quarter of 2016.

Top-10 and Bottom-10 along with the rents columns.

Top-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP	one_bed	two_bed	three_bed	four_plus_bed
Ku-ring-gai	11	11	1	2	109300	485	630	850	1200
The Hills Shire	23	22	2	2	169873	450	520	600	700
Lane Cove	2	3	4	6	31510	510	600	800	1275
Wollondilly	4	3	6	6	43261		350	425	550
Warringah	25	26	6	6	140740	470	610	850	1150
Hornsby	27	27	5	6	156849	430	500	630	780
Camden	5	7	8	8	56719	300	380	450	540
Pittwater	6	7	8	9	57154	450	620	850	1150
Ryde	20	20	10	9	103041	450	500	660	850
Hunters Hill	1	1	10	10	13217		575	900	1350

Bottom-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP	one_bed	two_bed	three_bed	four_plus_bed
Sydney	43	43	43	43	169507	550	750	1000	1200
Campbelltown	41	40	42	42	145970	295	350	410	500
Marrickville	30	31	41	40	76502	440	580	800	970
Parramatta	39	39	38	38	166859	400	455	530	655
Burwood	10	10	38	37	32424	445	550	650	915
Waverley	24	24	36	36	63485	600	770	1100	1600
Penrith	39	39	35	35	178466	250	340	420	540
Blacktown	42	42	36	34	301098	275	370	430	580
Botany Bay	13	12	34	33	39354	550	680	790	1120
Auburn	28	26	36	32	73738	450	520	550	655

Regional NSW - 110 LGAs

- Top-10 and Bottom-10 for the regional LGAs
- Sorted by "adj_mean_rank"

Top-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP
Unincorporated Far West	1	1	2	4	1168
Palerang	36	36	4	4	14351
Dungog	22	23	6	7	8316
Cabonne	36	36	7	7	12823
Gloucester	14	14	9	11	4879
Upper Lachlan Shire	24	24	12	11	7192
Boorowa	5	5	10	12	2399
Uralla	18	19	10	12	6032
Greater Hume Shire	32	33	10	13	9816
Coolamon	12	12	12	14	4100

Bottom-10	sum_median_rank	sum_mean_rank	adj_median_rank	adj_mean_rank	y2011POP
Bourke	70	68	110	110	2867
Brewarrina	42	44	109	109	1766
Central Darling	42	44	108	108	1992
Walgett	75	73	107	107	6453
Moree Plains	88	88	106	106	13428
Coonamble	52	52	105	105	4031
Narrandera	53	54	101	99	5900
Dubbo	102	102	100	98	38808
Byron	92	92	97	97	29207
Wellington	63	63	98	96	8494

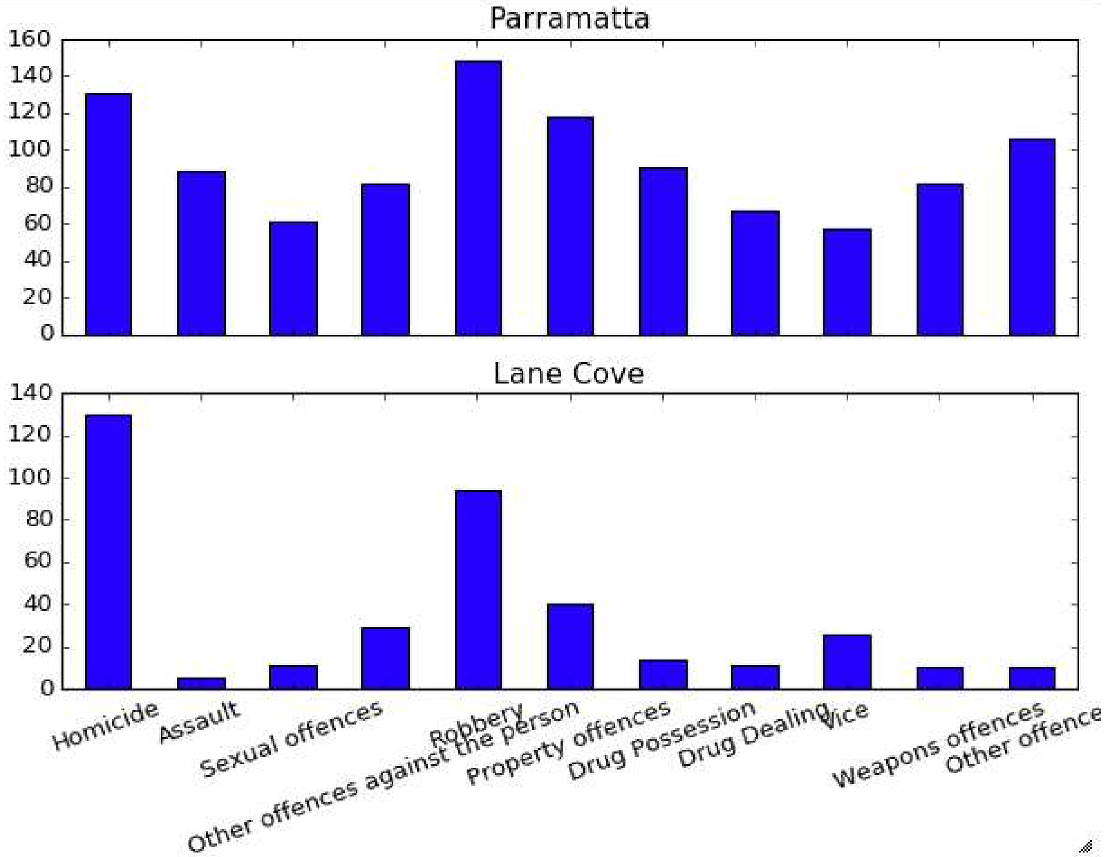
Method-2 analysis

- The below table shows sample output for some of the LGAs
- The values in each cell represent the mean rank for the categories
- Taking mean across each column gives the overall rank under method-2

Categories	Ashfield	Auburn	Blacktown	Byron	Coffs Harbour	Hunters Hill	Manly	Mosman	Parramatta	Ryde	Strathfield	Sydney
Homicide	134	131	130	131	132	134	134	131	131	129	136	143
Assault	42	65	107	111	104	13	71	7	89	16	43	147
Sexual offences	51	43	80	109	110	45	61	28	61	29	32	128
Other offences against the person	54	72	110	58	102	29	40	23	82	31	53	125
Robbery	141	147	138	91	108	107	110	94	148	115	147	153
Property offences	86	89	104	131	96	52	100	52	118	46	110	152
Drug Possession	69	108	75	151	122	24	79	20	91	31	48	150
Drug Dealing	56	78	52	140	106	22	61	16	67	27	39	143
Vice	40	44	53	139	96	34	117	62	57	34	25	144
Weapons offences	43	64	63	92	75	24	41	14	82	23	37	134
Other offences	72	107	100	122	86	13	84	13	106	13	47	144

- The below barplots show comparison of two LGAs

- Each bar represents the rank for that category. Smaller bars are better.



Comparison of ranking between method-1 and method-2 for LGAs in Greater Sydney

Method1-Top20 (Highest rank first)	Method2-Top20 (Highest rank first)	Method1-Bottom20 (Lowest rank first)	Method2-Bottom20 (Lowest rank first)
The Hills Shire	Ku-ring-gai	Sydney	Sydney
Ku-ring-gai	Lane Cove	Campbelltown	Campbelltown
Hornsby	The Hills Shire	Marrickville	Marrickville
Lane Cove	Mosman	Parramatta	Parramatta
Wollondilly	Hornsby	Burwood	Blacktown
Warringah	Canada Bay	Waverley	Penrith
Camden	Warringah	Penrith	Fairfield
Ryde	Pittwater	Blacktown	Auburn
Pittwater	Hunters Hill	Botany Bay	Waverley
Hunters Hill	Ryde	Auburn	Liverpool
Mosman	Camden	Manly	Botany Bay
Canada Bay	Willoughby	North Sydney	Bankstown
Sutherland Shire	Wollondilly	Liverpool	Manly
Blue Mountains	Woollahra	Leichhardt	Hawkesbury
Canterbury	Kogarah	Wyong	Burwood
Willoughby	North Sydney	Strathfield	Wyong
Kogarah	Sutherland Shire	Bankstown	Gosford
Rockdale	Hurstville	Fairfield	Holroyd
Hurstville	Canterbury	Ashfield	Ashfield
Randwick	Rockdale	Holroyd	Blue Mountains

Conclusion

Under Method-1 all crimes are given equal weightage, usually there will be lot more of petty crime compared to serious offences. For an LGA with a high count of petty crimes, their ranking will be pushed down under method-

1. An example of this is North Sydney, it moved from 32nd position under Method-1 to 23rd position under Method-2. Method-2 does not make this assumption. If there are high number of serious crimes in an LGA, it is reflected accordingly in the rankings.

Part 2

Instructions to run the notebook

Install if not already installed:

```
- jupyter      - pip3 install jupyter
- sqlalchemy   - pip3 install sqlalchemy --user unix_username
- psycopg2     - pip3 install psycopg2 --user unix_username
- ipywidgets
            - pip3 install ipywidgets --user unix_username
            - jupyter nbextension enable --py  widgetsnbextension
- nbconvert    - pip3 install nbconvert
```

- ipywidgets are needed for interactivity.
- sqlalchemy and psycopg2 are needed to interface with the postgres database
- nbconvert is needed to convert the notebook to HTML or Markdown, it uses pandoc (apt-get install pandoc)
    - eg: jupyter nbconvert stage2d.ipynb --to html

Change the postgres database connection variables (it's around 7 cells further down from the start of the Notebook)

Majority of the analysis was done with Python/Pandas, with Postgres database acting as storage backend for the cleaned data. Jupyter Notebook was used as a platform for the whole project. Jupyter ipywidgets were used to implement interactivity so users can interactively analyse the data. Jupyter nbconvert was used to generate the final reports from the Jupyter notebook.

First challenge was to clean the data and have it in a format that was conducive to analysis. I decided to load the data into Postgres SQL database so I could query the data/subsets of data whenever I needed it. Moving the data into postgres database was relatively straightforward with the powerful pandas dataFrame.to_sql() and sqlalchemy tools.

Setting up the persistent connection to the Postgres database:

First create a database:

```
postgres=# create database proj1903;
```

Then setup a persistent connection to the postgres database:

Change the below settings to connect to a Postgres database

```
PORT = "5432"          # this is the default PG port
LOGIN = "postgres"        # this is the user that has access to the database
PASSWORD = "password"      # user password
SERVER = "localhost"       # server where the database is located
DATABASE = "proj1903"      # database name

  engine = create_engine('postgresql://{LOGIN}:{PASSWORD}@{SERVER}:
  {PORT}/{DATABASE}'.format
```

```
(LOGIN=LOGIN,PASSWORD=PASSWORD, SERVER=SERVER, PORT=PORT, DATABASE=DATABASE))
```

Processing the data in NSWcrimes.csv

The idea is to store stats for each year in an SQL table.

- i.e, each table has:
  - 156 Attributes/Columns - 152 LGAs PLUS (Unincorporated Far West, Lord Howe Island, Prisons) and
  - 11 Tuples/Rows - 11 main categories

The data in NSWcrimes.csv has 64 categories/sub-categories, which will be reduced to the 11 main categories.

psudocode

- setup a defaultdict collection object template (temp_yr) that holds dictionaries and initialise it with LGA names as keys
- Define two more defaultdict collection objects to be used in the loop

- For each year:

  - open NSWcrimes.csv file using csv.dictreader
  - initialize a defaultdict object to hold the stats without grouping (i.e, for 62 categories), keys are LGAs
  - iterate over the reader dict object and save the stats to the dict of dicts (subs1)
  - initialize another defaultdict object to hold the grouped stats (subs_yr)
  - iterate over the subs1 dictionary and group the categories into 11 main categories
  - create a Pandas Dataframe form this final dictionary of dictionaries object (subs_yr)
  - write the dataframe to a SQL table

  Data was validated by comparing with the stats from the file RCI_offencebyyear.xlsx.

Most important snippets of code are:

The below defaultdict objects are a dictionary of dictionaries and are initialized with keys and no values.

subs1 is used to iterate over the file to read the data

Then subs_yr was used within the loop to reduce the categories from 64 to 11

```
subs1 = collections.defaultdict(dict)
subs_yr = collections.defaultdict(dict)
temp_yr = collections.defaultdict(dict)
for i in lga:
     temp_yr[i] = {}
```

Another interesting point was to realise that assigning a dictionary to another variable only creates a shallow copy and in the below case I needed a deep copy of the dictionary template, i.e,

only creates a reference to the temp_yr and not a true copy.

```
subs_yr = temp_yr
```

Below option is the right option, needs "import copy" at the top

```
subs_yr = copy.deepcopy(temp_yr)
```

At the end of each loop the dataFrame is written to an SQL table:

```
df1.to_sql(TABLE, engine, index_label='Categories')
```

This is only needed once, so needs to be commented out after the first successful run.

Rest of the code is straight forward and is commented in the main notebook file.

Interactivity

Interactivity throughout the notebook was provided by ipython widgets and interact() (or interact_manual()) functions.

The below code creates two widgets, one to select 1 or more LGAs and another to select the year from the dropdown list.

```
sel01 = widgets.SelectMultiple(   # this is useful for making multiple selections
    options=plga,              # These are the options
                              # presented to the user for selection
    value=['Hunters Hill', 'Blacktown', 'Ku-ring-gai', 'Lane Cove',
    'Leichhardt', 'Sydney'],    # Default selections
    description='LGA:',        # Description for the input field
    disabled=False,
    layout=Layout(display="inline_flex", flex_flow='column')
                              # to modify the displyed box
)

yer01  = widgets.Dropdown(       # this is useful for selecting just on option
    options=yrs,
    value='2016',
    description='Year:',
    disabled=False,
    layout=Layout(width='20%')
)
```

Issues with ipywidgets

Look at the below code:

```
def h01(s01, y01):
    global tab01            # declaring a global variable
                          #  to access the data outside this function
    y11 = 'y'+y01
    tab02 = pd.read_sql_table(y11,engine, index_col='Categories')
                        # reading the data from the postgres database
    tab01 = tab02[list(s01)]
    return(tab01)           # This shows the output but there
                            # was no way to capture the output
interact_manual(h01,  y01=yer01, s01=sel01)
```

Tried:

```
val = interact_manual(h01,  y01=yer01, s01=sel01)
```

But this does not save the returned data, instead saves the interact() function. After going through the ipywidgets documentation and not finding a suitable solution, resorted to using global variables or declaring the objects outside the interact function.

SQLAlchemy Library

SQLAlchemy was used to interface with the postgres database. Setting up the persistent connection was easy. It was building queries on the fly that was hard. SQLAlchemy has reasonably good documentation on how to use the expression language to query the database but they are mostly direct queries rather than building them from variables. There were quite a few examples on their website but I could not find many that relate to building queries in a loop using variables that change values with every loop.

I used SQLAlchemy queries when the where clause was not that complicated, even then it was quite challenging to build the query. At times it was easier to read the whole table into a pandas dataFrame and then slice the dataFrame to get the values needed than trying to build a query.

For example, below snippet was taken from the method-1 analysis:

I thought the below way of building the query was very clunky but it was hard to improve without understanding the "ORM" Queries. The below code is something I would have liked to improve given more time.

```
__USER SELECTION VALUES__
subs2 = sel9[lga9.value]
c1 = cat01.value
yr1 = yrs1.value

__GET THE DATA FROM THE POSTGRES DATABASE BASED ON THE USER SELECTION__

if len(c1) == 1:                  # IF ONLY ONE CATEGORY IS SELECTED
    c11 = str(list(c1)[0])
    for s1 in subs2:
        ps1 = int(pop1.loc['y2011POP',s1])
        for y1 in yr1:
            yr2 = 'y'+y1
            query='SELECT \"'+s1+'\" FROM '+ yr2+' WHERE "Categories"='+"'"+c11+"';"
            temp = pd.read_sql_query(query, engine)
            dsum.loc[y1,s1]=temp[s1].sum()
            dadj.loc[y1,s1]=round(((temp[s1].sum())/ps1)*10000)
else:
    c11 = str(c1)
    for s1 in subs2:
        ps1 = int(pop1.loc['y2011POP',s1])
        for y1 in yr1:
            yr2 = 'y'+y1
            query='SELECT \"'+s1+'\" FROM '+ yr2+' WHERE "Categories" IN'+c11+';'
            temp = pd.read_sql_query(query, engine)
            dsum.loc[y1,s1]=temp[s1].sum()
            dadj.loc[y1,s1]=round(((temp[s1].sum())/ps1)*10000)
                                  # population adjusted per 10000 people
```

Pandas

Heavy lifting was done by Pandas DataFrame.

They were used for ranking, plotting, and writing and retrieving data from Postgres database and numerous other data manipulations.

Some of the interesting code snippets are:

To save a dataframe to an SQL database:

```
df1.to_sql(TABLE, engine, index_label='Categories')
```

Reading a table form an SQL database:

```
pd.read_sql_table(y11,engine, index_col='Categories')
```

Dividing a dataframe by another

```
pop01 = np.array(pop1[list(sel01.value)])      # building a 2D-array of population counts
pop02 = np.tile(pop01, (11,1))                 # with the first row repeating 11 times
pdiv = pop02.astype('int')
tadj01 = (tab01*10000).div(pdiv, axis='columns').astype('int')
        # dividing a dataframe by another
```

Plotting with a dataFrame:

```
tadj01.T.plot.bar(stacked=True, ax=ax21, legend=False,
  title='Population Adjusted', colormap='Set1')
tab01.T.plot.bar(stacked=True, ax=ax22, legend=False,
  title='Not Adjusted', rot=40, colormap='Set1')
```

df.rank() was used several times to get the ranks

```
dadjr_men = dadj.rank(axis=1,method='max',
ascending=True).mean(axis=0).sort_values().to_frame()
```

Below code gives the overall mean for the LGAs

```
altrank = final3.mean(axis=0).round().astype(int).sort_values().to_frame('Mean Rank')
```

References

Files provided along with this file and the Jupyter Notebook

File Names	Comments
2011LGA_pop.csv	Population counts **(Used in this notebook)**
lga.txt	Names of all LGAs **(Used in this notebook)**
NSWcrimes.csv	Main data set **(Used in this notebook)**
offence_categories.xlsx	Offence categories and sub categories
RCI_offencebyyear.xlsx	Used for data validation
rents_gmr.csv	Rents data set **(Used in this notebook)**
source_files folder	Has unmodified source files

source_files folder	Has unmodified source files
LGApop2011.xlsx	Population counts downloaded from ABS

source_files folder	Has unmodified source files
OffenceCategories-2014.pdf	Document explaining the mapping of police crime categories to BOCSAR crime categories
RCI_offencebymonth.xlsm	Original/main data set of crime stats
Rent_Report_16q4.xls	Original rents data set

RCI_offencebymonth.xlsm can be downloaded from:

http://www.bocsar.nsw.gov.au/Documents/RCS-Annual/RCI_offencebymonth.zip

Rent_Report_16q4.xls can be downloaded from:

http://www.housing.nsw.gov.au/__data/assets/excel_doc/0003/408828/Rent_Report_16q4.xls