DMHR Assignment

Part A: NHS Prescribing

Unnecessary prescriptions increase waste, are harmful to patients and are costly to the NHS, as well as contributing to issues such as antimicrobial resistance. One of the first steps in reducing unnecessary prescriptions at a large scale is to examine patterns in prescribing data across GPs, where most prescription occurs. This report explores prescription in London and Cambridge practices in April 2018 before exploring national prescribing in all practices for cardiovascular drugs and antidepressant drugs. Finally, prescriptions across England are explored. Variation in prescribing practice could be used to inform policy to improve practice and increase efficiency.

Set up & background

Data for prescribing by practice was taken from NHS Digital and combined with NHS Digital GPs population address details.

In [1]:
```
# import modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
import re
```

In [2]:
```
# import addresses
colnames = ["DATE", "CODE", "NAME", "ADD1", "ADD2", "ADD3", "ADD4", "POSTCODE"]
add_raw = pd.read_csv("T201804ADDR+BNFT.csv", header = None, names = colnames)
add_raw = add_raw.apply(lambda x: x.str.strip() if x.dtype == "object" else x)

# filter to address data to London & Cambridge only
## some practices contain the city in ADD4 while others have it in ADD3 so both have bee
n used
## exact matching has been used in most cases to ensure insitsnces such as 'London Road'
are not returned
## further details can be found in the sections for the respective areas
add_ldn = add_raw[add_raw['ADD4'].str.contains('LONDON') | (add_raw['ADD3'] == 'LONDON')
]
add_cbg = add_raw.loc[(add_raw['ADD3'] == 'CAMBRIDGE') | (add_raw['ADD4'] == 'CAMBRIDGE'
)]
```

In [3]:
```
# import prescribing data
colnames = ['SHA', 'PCT', 'CODE', 'BNFCODE', 'BNFNAME', 'ITEMS', 'NIC', 'ACTCOST', 'QUAN
TITY', 'PERIOD']
scrips_raw = pd.read_csv('T201804PDPI+BNFT.csv', header = 0, names = colnames, index_col
=False)
pop_raw = pd.read_csv('gp-reg-pat-prac-all.csv')
```

In [4]:
```
# add total cost column
scrips_raw.loc[:,'TOTALCOST'] = scrips_raw['ITEMS'] * scrips_raw['ACTCOST']
```

Some records in the prescription data are for 'dummy' practices created to account for prescriptions in primary care outside of general practices (e.g. prisons, hospices etc.). This leads to a mismatch in the number of providers when compared to other data sets such as population. The population data will be used as the definitive list of active, primary care general practices and the address data will be used to subset this into a list of codes for the areas of interest. Because branch practices may be recorded differently in different datasets it is possible that mismatches will still occur.

In [5]:
```
# define columns of interest
cols = ['CODE', 'NAME', 'NUMBER_OF_PATIENTS', 'BNFCODE', 'BNFNAME', 'ITEMS', 'ACTCOST',
'QUANTITY', 'TOTALCOST']

# get London data
ldn = pd.merge(pd.merge(add_ldn, scrips_raw, on='CODE', how='inner'), pop_raw, on='CODE'
, how = 'inner')
ldn = ldn[cols]

# get Cambridge data
cbg = pd.merge(pd.merge(add_cbg, scrips_raw, on='CODE', how='inner'), pop_raw, on='CODE'
, how = 'inner')
cbg = cbg[cols]
```

London

London practices are identified as those with:

- London (exact match) in the third address attribute ('add3') because some records do not have data in the fourth address column. An exact match is used to avoid including practices that are located on roads such as 'London Road' which may not be in London.
- London contained in the fourth address attribute ('add4') because some records hold both the area of London and London in this attribute (e.g. 'FINCHLEY LONDON')

It is possible that some London practices have been unintentionally excluded or non-London practices included however by using relatively restrictive rules this risk is minimised to the extent that the data allows.

The list of practices was further reduced to include only those where population data for April 2018 was present. This ensures consistency between analyses as well as helping reduce the number of non-general practices in the prescribing data.

In [6]:
```
# details of matching
print('While', add_ldn['CODE'].nunique() ,'practices where identified from the address l
ookup as in London, only', ldn['CODE'].nunique(),
      'of these where present in the population data. This may reflect practices in the
address data that are dummy practices, not general practices, have closed or are dormant
or practices that are missing population data.')
```

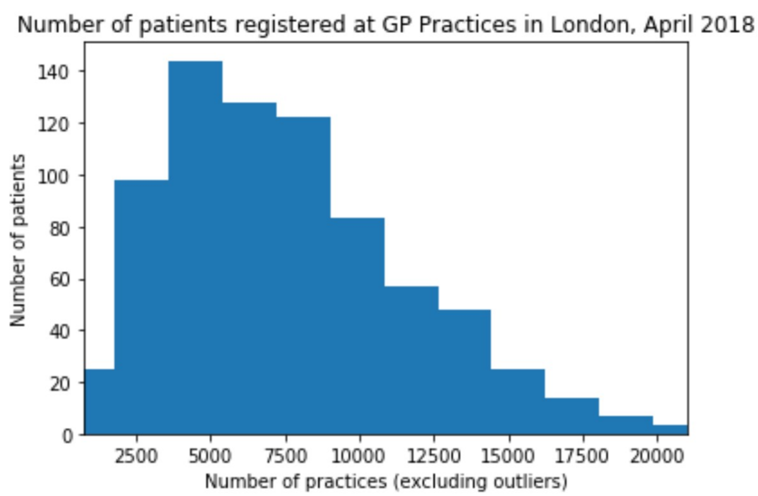
```
While 943 practices where identified from the address lookup as in London, only 761 of
these where present in the population data. This may reflect practices in the address
data that are dummy practices, not general practices, have closed or are dormant or pr
actices that are missing population data.
```

In [7]:
```
# summary of London GP pop
print('The total number of patients registered at the ', ldn['CODE'].nunique(),
      ' GP practices in London during April 2018 was ', ldn.groupby('CODE')['NUMBER_OF_P
ATIENTS'].first().sum(),
      '. The average practice had ', round(ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].fir
st().mean()),
      ' patients registered but this is skewed by a few large practices.',
      ' The median number of patients registered per practice is ', ldn.groupby('CODE')[
'NUMBER_OF_PATIENTS'].first().median(),
      '. As can be seen in the following summary statistics and histogram the range betw
een practices is quite large.'
      , sep='')

ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first().describe()
```

```
The total number of patients registered at the 761 GP practices in London during April
2018 was 5959862. The average practice had 7832 patients registered but this is skewed
by a few large practices. The median number of patients registered per practice is 705
0.0. As can be seen in the following summary statistics and histogram the range betwee
n practices is quite large.
```

Out[7]:
```
count      761.000000
mean      7831.618922
std       5030.513284
min          8.000000
25%       4570.000000
50%       7050.000000
75%      10225.000000
max      72227.000000
Name: NUMBER_OF_PATIENTS, dtype: float64
```

In [8]:
```
# hist of London pop, exlcuding 0.01 outliers
min_x = ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first().quantile(.01)
max_x = ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first().quantile(.99)
_ = plt.hist(ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first(), bins = 40)
_ = plt.xlim(min_x, max_x)
_ = plt.title('Number of patients registered at GP Practices in London, April 2018')
_ = plt.xlabel('Number of practices (excluding outliers)')
_ = plt.ylabel('Number of patients')
```



There is a marked positive skew to the histogram, suggesting few small practices but more very large practices.

Prescriptions

In [9]:
```
# group prescribing data by practice
ldn_prac_grouped = ldn.drop(['ACTCOST'], axis=1).groupby(['CODE', 'NAME', 'NUMBER_OF_PAT
IENTS']).sum().reset_index('NUMBER_OF_PATIENTS')

# group prescribing data by BNFCODE
ldn_bnf_grouped = ldn.drop(['NUMBER_OF_PATIENTS', 'ACTCOST'], axis=1).groupby(['BNFCODE'
, 'BNFNAME']).sum().reset_index('BNFCODE')
```

In [10]:
```
# summary of London GP scrips
print('Within London ', np.sum(ldn['ITEMS']), ' prescriptions for ', ldn['BNFCODE'].nuni
que(), ' different items were given out in April 2018 ',
      'at a total cost of £', round(np.sum(ldn['TOTALCOST']),2), '. This means across Lo
ndon the average number of prescriptions per patient was ',
      round(np.sum(ldn['ITEMS']) / ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first().sum
(), 2), '. On average each patient cost £',
      round(np.sum(ldn['TOTALCOST']) / ldn.groupby('CODE')['NUMBER_OF_PATIENTS'].first()
.sum(),2), ' with each costing on average £',
      round(np.mean(ldn['ACTCOST']),2), '.',
      sep = '')

ldn.groupby('CODE').first().describe()
```

```
Within London 5941115 prescriptions for 13054 different items were given out in April
2018 at a total cost of £1022776544.92. This means across London the average number of
prescriptions per patient was 1.0. On average each patient cost £171.61 with each cost
ing on average £54.26.
```

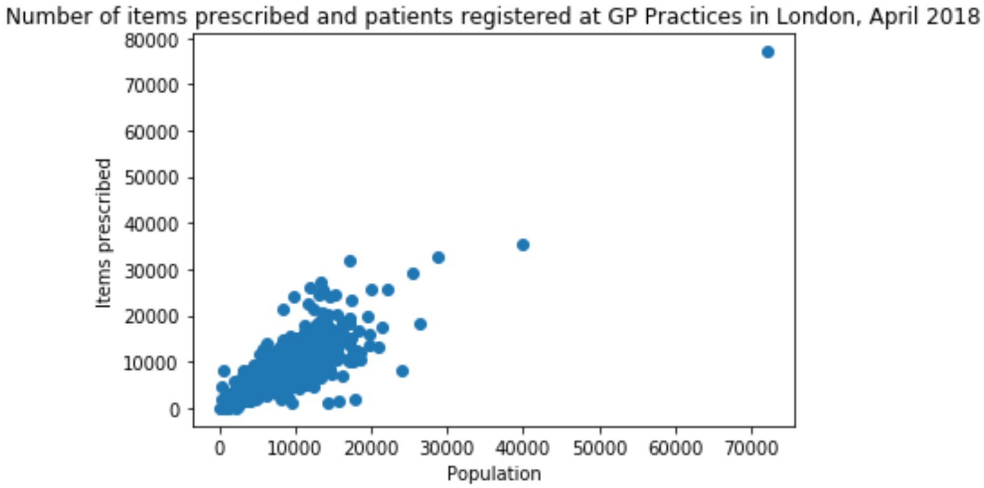
Out[10]:

	NUMBER_OF_PATIENTS	ITEMS	ACTCOST	QUANTITY	TOTALCOST
count	761.000000	761.000000	761.000000	761.000000	761.000000
mean	7831.618922	2.864652	24.386413	1331.244415	133.508239
std	5030.513284	4.076418	63.351682	2213.847171	459.503700
min	8.000000	1.000000	0.290000	14.000000	0.290000
25%	4570.000000	1.000000	2.790000	224.000000	2.790000
50%	7050.000000	1.000000	6.460000	500.000000	10.100000
75%	10225.000000	3.000000	18.700000	1300.000000	51.920000
max	72227.000000	32.000000	739.120000	16350.000000	5007.040000

When looking at the average of practices (as opposed to the population overall) we can see that there was large range in the average number and total cost of prescriptions across the practices.

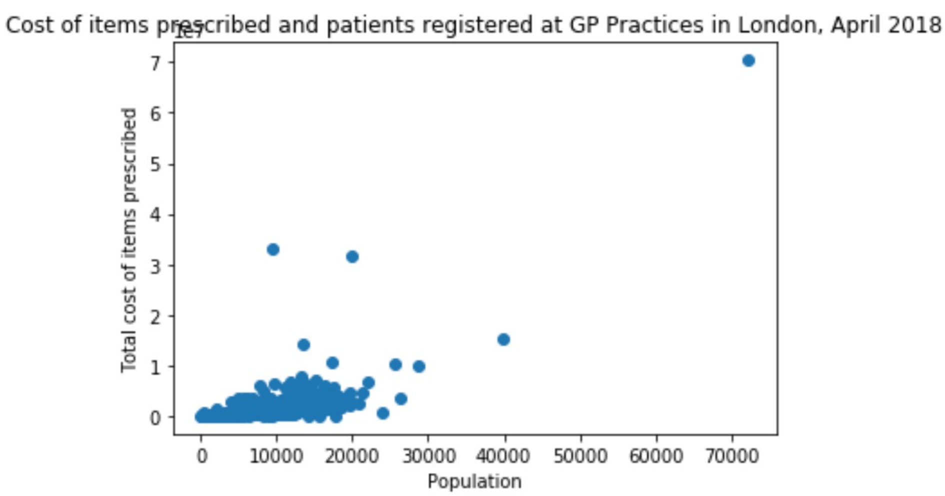
There is considerable variation between practices in the number of items prescribed and the cost of the items prescribed. There could be several reasons for this such as number of patients and type of medications prescribed. To explore these differences we can look at prescriptions by the registered population.

In [11]:
```
# scatter of pop vs scrips
_ = plt.scatter(ldn_prac_grouped['NUMBER_OF_PATIENTS'], ldn_prac_grouped['ITEMS'])
_ = plt.title('Number of items prescribed and patients registered at GP Practices in Lon
don, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Items prescribed')
```



This suggests much of the variation in the number of items prescribed is due to the number of patients

In [12]:
```
# scatter of pop vs scrip cost
_ = plt.scatter(ldn_prac_grouped['NUMBER_OF_PATIENTS'], ldn_prac_grouped['TOTALCOST'])
_ = plt.title('Cost of items prescribed and patients registered at GP Practices in Londo
n, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Total cost of items prescribed')
```



However, the cost of prescriptions appears to be less correlated with practice size suggesting there may be differences in prescribing practice between GPs or differences in the needs of patients between practices, for example a tendency to prescribe more expensive items and (as opposed to simply more) would increase costs. Work could be undertaken to explore if more cost-effective items could be substituted for more costly items. There may be particular items that account for much of the expenditure.

In [13]:
```python
# most commly prescribed
print('The 10 most prescribed items were:')
ldn_bnf_grouped.sort_values('ITEMS', axis=0, ascending=False, kind='quicksort', na_posit
ion='last').head(10)
```

```
The 10 most prescribed items were:
```

Out[13]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Omeprazole_Cap E/C 20mg	0103050P0AAAAAA	125415	4752282	26772684.61
Metformin HCl_Tab 500mg	0601022B0AAABAB	123250	11930509	97133900.72
Amlodipine_Tab 5mg	0206020A0AAAAAA	105391	3807839	37651119.13
Atorvastatin_Tab 20mg	0212000B0AAABAB	101891	3645805	19203632.58
Amlodipine_Tab 10mg	0206020A0AAABAB	97798	3400459	38024034.00
Atorvastatin_Tab 40mg	0212000B0AAACAC	92801	2702776	18814193.46
Aspirin Disper_Tab 75mg	0209000A0AAABAB	89837	2480942	6275011.09
Lansoprazole_Cap 30mg (E/C Gran)	0103050L0AAAAAA	89570	2570061	16989100.52
Salbutamol_Inha 100mcg (200 D) CFF	0301011R0AAAPAP	81474	116861	28544957.38
Paracet_Tab 500mg	0407010H0AAAMAM	75805	7850343	13036136.69

In [14]:
```python
# most costly
print('While the ten most costly items were:')
ldn_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=False, kind='quicksort', na_p
osition='last').head(10)
```

```
While the ten most costly items were:
```

Out[14]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Metformin HCl_Tab 500mg	0601022B0AAABAB	123250	11930509	97133900.72
Influenza_Vac SplitViron Inact 0.5ml Pfs	1404000H0AAAKAK	8400	8400	64104457.45
Sitagliptin_Tab 100mg	0601023X0AAAAAA	27320	799463	62629922.71
Amlodipine_Tab 10mg	0206020A0AAABAB	97798	3400459	38024034.00
Amlodipine_Tab 5mg	0206020A0AAAAAA	105391	3807839	37651119.13
Salbutamol_Inha 100mcg (200 D) CFF	0301011R0AAAPAP	81474	116861	28544957.38
Omeprazole_Cap E/C 20mg	0103050P0AAAAAA	125415	4752282	26772684.61
Atorvastatin_Tab 20mg	0212000B0AAABAB	101891	3645805	19203632.58
Atorvastatin_Tab 40mg	0212000B0AAACAC	92801	2702776	18814193.46
Rivaroxaban_Tab 20mg	0208020Y0AAACAC	12346	357519	18057412.71

In [15]:
```
# least commonly prescribed
print('The 10 least prescribed items were (note',
      len(ldn_bnf_grouped.loc[ldn_bnf_grouped['ITEMS'] == 1]), 'items are tied for 1):')
ldn_bnf_grouped.sort_values('ITEMS', axis=0, ascending=True, kind='quicksort', na_positi
on='last').head(10)
```

```
The 10 least prescribed items were (note 2558 items are tied for 1):
```

Out[15]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Peak_Uromate Urost Pouch Transpt Lge S/H	23965909618	1	30	174.55
Welland_FreeStyle Vie Closed Pouch Clr L	23355603799	1	120	329.89
Welland_FreeStyle Vie Closed Pouch Clr L	23355603797	1	60	164.95
Welland_FreeStyle Vie Closed Pouch Clr L	23355603796	1	90	247.42
Psytixol_Inj 100mg/ml 0.5ml Amp	0402020G0BCACAH	1	10	31.66
ActiLymph Class 3 B/Knee Open Toe Wt Top	21270000170	1	2	29.43
Welland_FreeStyle Vie Closed Pouch Clr M	23355603794	1	60	164.95
Welland_FreeStyle Vie Closed Pouch Beige	23355603792	1	120	329.89
Modecate Conc_Inj 100mg/ml 0.5ml Amp	0402020L0BBAFAB	1	1	4.15
Welland_FreeStyle Vie Closed Pouch Clr L	23355603800	1	90	247.42

In [16]:
```
# least costly
print('While the ten least costly items were:')
ldn_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=True, kind='quicksort', na_po
sition='last').head(10)
```

```
While the ten least costly items were:
```

Out[16]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
365 Film 4cm x 5cm VP Adh Film Dress	20030100662	1	2	0.09
Caffeine_Tab 50mg	0404000D0AAAAAA	1	1	0.17
Ritalin SR_Tab 20mg (Import)	0404000M0BBABAH	1	120	0.18
Chemipore 1.25cm x 5m Surg Adh Tape Perm	20100000565	1	1	0.26
Lloyds_Multivit Tab	090607000BBGYA0	1	28	0.33
Unspecified Pfa 5cm x 5cm Plas Faced Dre	20030100100	1	14	0.40
SurgiSense_Uridrop Sheath Size 2	22304753001	1	1	0.50
Cod Liver Oil_Cap 525mg	0906011H0AABBBB	1	7	0.57
Adpore Ultra 10cm x 10cm VP Adh Film Dre	20030100616	1	5	0.66
Suspen For Mens Stkn	21070100990	1	1	0.68

There is likely to be substantial cost in keeping medications available even if they are not routinely used. It may be possible to stop or reduce the supply of less prescribed items to reduce cost without affect clinician or patient choice by encouraging the use of alternative products or different pack sizes.

Cambridge

Cambridge practices are identified as those with:

- Cambridge (exact match) in the third address attribute ('add3') because some records do not have data in the fourth address column. An exact match is used to avoid including practices that are located on roads such as 'Cambridge Road' which may not be in Cambridge.
- Cambridge (exact match) in the fourth address attribute ('add4') because some records do not have data in the fourth address column. An exact match is used to avoid including practices that are located on roads such as 'Cambridge Road' which may not be in Cambridge.

It is possible that some Cambridge practices have been unintentionally excluded or non-Cambridge practices included however by using relatively restrictive rules this risk is minimised to the extent that the data allows.

The list of practices was further reduced to include only those where population data for April 2018 was present. This ensures consistency between analyses as well as helping reduce the number of non-general practices in the prescribing data.

In [17]:
```
# caveats of Cambridge pop
print('While', add_cbg['CODE'].nunique() ,'practices where identified from the address l
ookup as in Cambridge, only ', cbg['CODE'].nunique(),
      'of these where present in the population data. This may reflect practices in the
address data that are dummy practices, not general practices, have closed or are dormant
or practices that are missing population data.')
```

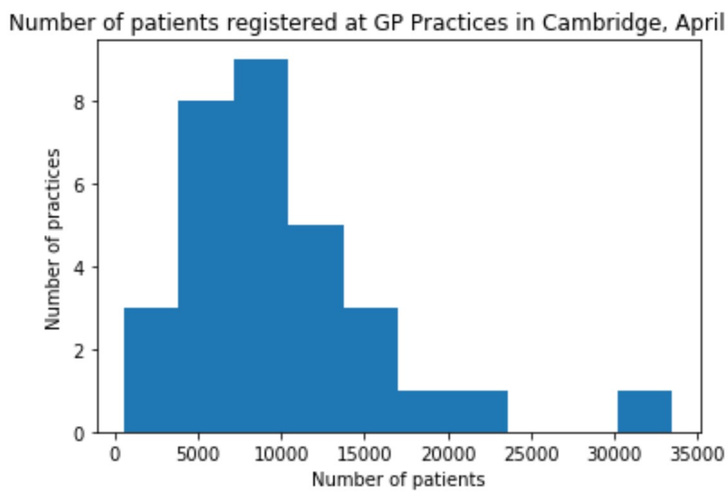
```
While 36 practices where identified from the address lookup as in Cambridge, only  31
of these where present in the population data. This may reflect practices in the addre
ss data that are dummy practices, not general practices, have closed or are dormant or
practices that are missing population data.
```

In [18]:
```
# summary of GP pop
print('The total number of patients registered at the ', cbg['CODE'].nunique(),
      ' GP practices in Cambridge during April 2018 was ', cbg.groupby('CODE')['NUMBER_O
F_PATIENTS'].first().sum(),
      '. The average practice had ', round(cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].fir
st().mean()),
      ' patients registered but this is skewed by a few large practices.',
      ' The median number of patients registered per practice is ', cbg.groupby('CODE')[
'NUMBER_OF_PATIENTS'].first().median(),
      '. As can be seen in the following summary statistics and histogram the range betw
een practices is quite large.'
      , sep='')

cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first().describe()
```

```
The total number of patients registered at the 31 GP practices in Cambridge during Apr
il 2018 was 311579. The average practice had 10051 patients registered but this is ske
wed by a few large practices. The median number of patients registered per practice is
9063.0. As can be seen in the following summary statistics and histogram the range bet
ween practices is quite large.
```

Out[18]:
```
count       31.000000
mean     10050.935484
std       6275.995708
min        568.000000
25%       6099.500000
50%       9063.000000
75%      12172.500000
max      33501.000000
Name: NUMBER_OF_PATIENTS, dtype: float64
```

In [19]:
```
#hist of Cambridge pop
#min_x = cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first().quantile(.01)
#max_x = cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first().quantile(.99)
_ = plt.hist(cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first())
#_ = plt.xlim(min_x, max_x)
_ = plt.title('Number of patients registered at GP Practices in Cambridge, April 2018')
_ = plt.xlabel('Number of patients')
_ = plt.ylabel('Number of practices')
```



There is a marked positive skew to the histogram, suggesting few small practices but more very large practices.

Prescriptions

In [20]:
```
# group prescribing data by practice
cbg_prac_grouped = cbg.drop(['ACTCOST'], axis=1).groupby(['CODE', 'NAME', 'NUMBER_OF_PAT
IENTS']).sum().reset_index('NUMBER_OF_PATIENTS')

# group prescribing data by BNFCODE
cbg_bnf_grouped = cbg.drop(['NUMBER_OF_PATIENTS', 'ACTCOST'], axis=1).groupby(['BNFCODE'
, 'BNFNAME']).sum().reset_index('BNFCODE')
```

In [21]:
```
# summary of Cambridge GP scrips
print('Within Cambridge ', np.sum(cbg['ITEMS']), ' prescriptions for ', cbg['BNFCODE'].n
unique(), ' different items were given out in April 2018 ',
      'at a total cost of £', round(np.sum(cbg['TOTALCOST']),2), '. This means across Ca
mbridge the average number of prescriptions per patient was ',
      round(np.sum(cbg['ITEMS']) / cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first().sum
(), 2), '. On average each patient cost £',
      round(np.sum(cbg['TOTALCOST']) / cbg.groupby('CODE')['NUMBER_OF_PATIENTS'].first()
.sum(),2), ' with each drug costing on average £',
      round(np.mean(cbg['ACTCOST']),2),
      sep = '')

cbg.groupby('CODE').first().describe()
```

```
Within Cambridge 344645 prescriptions for 5683 different items were given out in April
2018 at a total cost of £67370054.56. This means across Cambridge the average number o
f prescriptions per patient was 1.11. On average each patient cost £216.22 with each d
rug costing on average £60.73
```

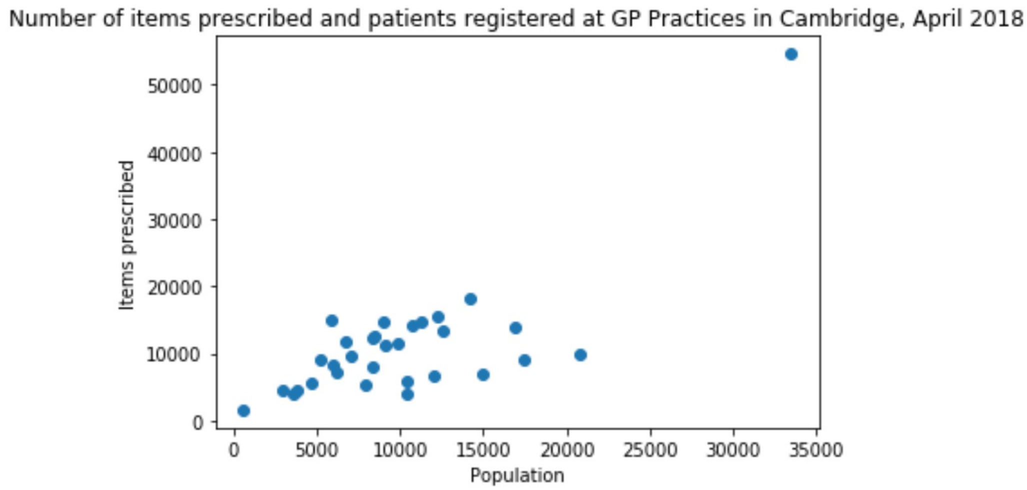
Out[21]:

	NUMBER_OF_PATIENTS	ITEMS	ACTCOST	QUANTITY	TOTALCOST
count	31.000000	31.000000	31.000000	31.000000	31.000000
mean	10050.935484	2.161290	16.039355	1191.161290	48.588387
std	6275.995708	2.146265	20.782409	1743.296095	96.847555
min	568.000000	1.000000	1.120000	24.000000	1.820000
25%	6099.500000	1.000000	2.955000	275.000000	2.955000
50%	9063.000000	1.000000	5.980000	500.000000	7.580000
75%	12172.500000	2.500000	18.440000	1075.000000	35.840000
max	33501.000000	9.000000	85.510000	8110.000000	383.130000

When looking at the average of practices (as opposed to the population overall) we can see that there was large range in the average number and total cost of prescriptions across the practices.

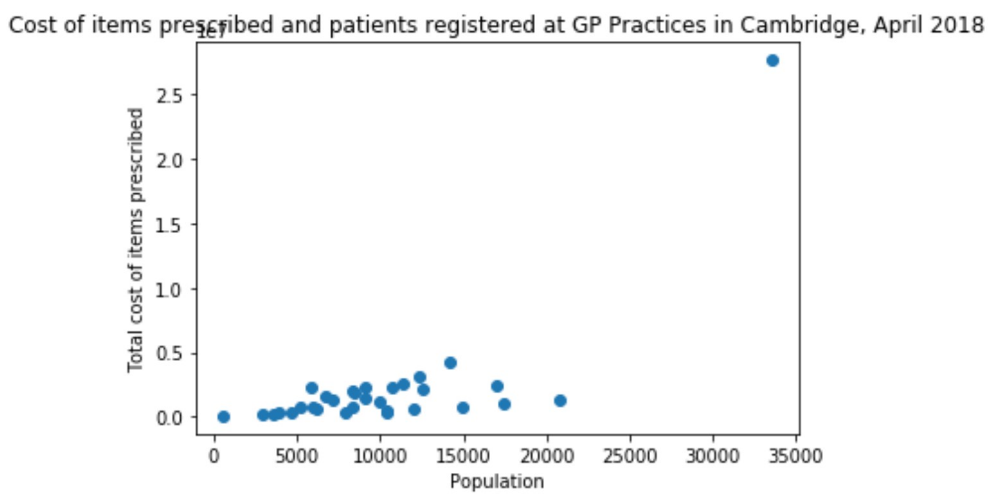
There is considerable variation between practices in the number of items prescribed and the cost of the items prescribed. There could be several reasons for this such as number of patients and type of medications prescribed. To explore these differences we can look at prescriptions by the registered population.

In [22]:
```
# scatter of pop vs scrips
_ = plt.scatter(cbg_prac_grouped['NUMBER_OF_PATIENTS'], cbg_prac_grouped['ITEMS'])
_ = plt.title('Number of items prescribed and patients registered at GP Practices in Cam
bridge, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Items prescribed')
```



This suggests much of the variation in the number of items prescribed is due to the number of patients

In [23]:
```
# scatter of pop vs scrip cost
_ = plt.scatter(cbg_prac_grouped['NUMBER_OF_PATIENTS'], cbg_prac_grouped['TOTALCOST'])
_ = plt.title('Cost of items prescribed and patients registered at GP Practices in Cambr
idge, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Total cost of items prescribed')
```



However, the cost of prescriptions appears to be less correlated with practice size suggesting there may be differences in prescribing practice between GPs or differences in the needs of patients between practices, for example a tendency to prescribe more expensive items and (as opposed to simple more) would increase costs. Work could be undertaken to explore if more cost-effective items could be substituted for more costly items. There may be particular items that account for much of the expenditure.

In [24]:
```
# most commonly prescribed
print('The 10 most prescribed items were:')
cbg_bnf_grouped.sort_values('ITEMS', axis=0, ascending=False, kind='quicksort', na_posit
ion='last').head(10)
```

```
The 10 most prescribed items were:
```

Out[24]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Omeprazole_Cap E/C 20mg	0103050P0AAAAAA	11675	443377	6144577.43
Atorvastatin_Tab 20mg	0212000B0AAABAB	6416	208712	1756541.61
Aspirin Disper_Tab 75mg	0209000A0AAABAB	5734	170129	570786.75
Amlodipine_Tab 5mg	0206020A0AAAAAA	5462	176665	2579546.05
Paracet_Tab 500mg	0407010H0AAAMAM	4564	573883	1467760.93
Salbutamol_Inha 100mcg (200 D) CFF	0301011R0AAAPAP	4327	5279	1495281.02
Levothyrox Sod_Tab 100mcg	0602010V0AABZBZ	3895	137108	968612.88
Amitriptyline HCl_Tab 10mg	0403010B0AAAGAG	3849	196834	1397160.88
Simvastatin_Tab 40mg	0212000Y0AAADAD	3795	113735	686841.13
Levothyrox Sod_Tab 25mcg	0602010V0AABWBW	3793	134374	1652622.15

In [25]:
```
# most costly
print('While the ten most costly items were:')
cbg_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=False, kind='quicksort', na_p
osition='last').head(10)
```

```
While the ten most costly items were:
```

Out[25]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Omeprazole_Cap E/C 20mg	0103050P0AAAAAA	11675	443377	6144577.43
Fostair_Inh 100mcg/6mcg (120D) CFF	0302000C0BQAABX	1725	2046	5382211.81
Rivaroxaban_Tab 20mg	0208020Y0AAACAC	1177	31530	3534473.06
Amlodipine_Tab 5mg	0206020A0AAAAAA	5462	176665	2579546.05
Tamsulosin HCl_Cap 400mcg M/R	0704010U0AAAAAA	2828	89515	1840208.75
Atorvastatin_Tab 20mg	0212000B0AAABAB	6416	208712	1756541.61
Levothyrox Sod_Tab 25mcg	0602010V0AABWBW	3793	134374	1652622.15
Metformin HCl_Tab 500mg	0601022B0AAABAB	3620	328345	1633485.62
Salbutamol_Inha 100mcg (200 D) CFF	0301011R0AAAPAP	4327	5279	1495281.02
Citalopram Hydrob_Tab 20mg	0403030D0AAAAAA	3326	114927	1469082.46

In [26]:
```
print('The 10 least prescribed items were (note',
      len(cbg_bnf_grouped.loc[cbg_bnf_grouped['ITEMS'] == 1]), 'items are tied for 1):')
cbg_bnf_grouped.sort_values('ITEMS', axis=0, ascending=True, kind='quicksort', na_positi
on='last').head(10)
```

```
The 10 least prescribed items were (note 1524 items are tied for 1):
```

Out[26]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Mag Carb_Heavy Cap 500mg	0101010F0AAAUAU	1	30	85.51
Resource_2.0 Fibre Liq Feed (6 Flav)	090402000BBNIA0	1	20736	405.10
Jevity 1.5kcal_Liq	090402000BBNJA0	1	30000	338.88
Jevity Promote_Liq	090402000BBNVA0	1	28000	293.18
Ensure TwoCal_Liq (4 Flav)	090402000BBRZA0	1	30000	308.83
Fresubin Thickened Stage 1_Syr (2 Flav)	090402000BBSUA0	1	800	8.72
Peptamen Junior Advance_Liq	090402000BBTDA0	1	28000	418.08
Fresubin Pdr Extra_Pdr Sach 62g (Sbery)	090402000BBUIA0	1	28	18.19
Nutrison Pack_Protein Plus M/Fibre	090402000BBNDA0	1	28000	313.69
Fresubin Pdr Extra_Pdr Sach 62g(Vanilla)	090402000BBUJA0	1	14	9.09

In [27]:
```
# least costly
print('While the ten least costly items were:')
cbg_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=True, kind='quicksort', na_po
sition='last').head(10)
```

```
While the ten least costly items were:
```

Out[27]:

	BNFCODE	ITEMS	QUANTITY	TOTALCOST
BNFNAME				
Haddenham Acc For Veni MTO Short Leg	21270002508	1	4	0.00
Haddenham Acc For Veni MTO Short Foot	21270002510	1	4	0.00
Haddenham Acc For Veni MTO Non-Stnd Colr	21270002513	1	4	0.00
Haddenham Acc For Veni MTO 5cm Strong Pl	21270002515	1	4	0.00
K-Band 10cm x 4m Ktd Polyam & Cellulose	20020200803	1	1	0.27
Sure-Amp_Lido HCl Inj 2% 5ml Amp	1502010J0BLABCB	1	1	0.28
Mepore 11cm x 15cm Pfa + Adh Border Dres	20030100079	1	1	0.33
Jobst Elvarex Acc For L/Extrem Closed To	21270000118	1	4	0.41
Unspecified Surg Adh Tape 2.5cm x 5m Per	20100000552	1	1	0.42
Olive Oil_Liq	190605000AACACA	2	20	0.43

There is likely to be substantial cost in keeping medications available even if they are not routinely used. It may be possible to stop or reduce the supply of less prescribed items to reduce cost without affecting clinician or patient choice by encouraging the use of alternative products or different pack sizes.

London compared to Cambridge

Differences in prescribing items, quantities and costs may present opportunities for efficiency savings or may suggest that different approaches are needed to tackle the issue in different areas.

In [28]:

```
# summary of Cambridge GP scrips
print('Within Cambridge ', np.sum(cbg['ITEMS']), ' prescriptions for ', cbg['BNFCODE'].n
unique(), ' different items were given out in April 2018 ',
      'at a total cost of £', round(np.sum(cbg['TOTALCOST']),2), '. This means across Ca
mbridge the average number of prescriptions per patient was ',
      round(np.sum(cbg['ITEMS']) / np.sum(cbg_prac_grouped['NUMBER_OF_PATIENTS']), 2), '
. On average each patient cost £',
      round(np.sum(cbg['TOTALCOST']) / np.sum(cbg_prac_grouped['NUMBER_OF_PATIENTS']),2)
, ' with each item costing on average £',
      round(np.mean(cbg['ACTCOST']),2),
      sep = '')

cbg.describe()
```

```
Within Cambridge 344645 prescriptions for 5683 different items were given out in April
2018 at a total cost of £67370054.56. This means across Cambridge the average number o
f prescriptions per patient was 1.11. On average each patient cost £216.22 with each i
tem costing on average £60.73
```

Out[28]:

	NUMBER_OF_PATIENTS	ITEMS	ACTCOST	QUANTITY	TOTALCOST
count	39998.000000	39998.000000	39998.000000	39998.000000	3.999800e+04
mean	11617.116856	8.616556	60.726206	629.808415	1.684336e+03
std	7231.055223	29.363010	157.451384	3058.436264	2.638299e+04
min	568.000000	1.000000	0.000000	0.000000	0.000000e+00
25%	7115.000000	1.000000	7.420000	28.000000	1.176000e+01
50%	9927.000000	2.000000	20.160000	90.000000	4.730000e+01
75%	14231.000000	6.000000	54.410000	308.000000	2.429600e+02
max	33501.000000	2087.000000	8099.100000	159072.000000	3.206613e+06

In [29]:
```
# pop differences
print('As would be expected the registered population of London is substantially larger
than Cambridge (by ',
      np.sum(ldn_prac_grouped['NUMBER_OF_PATIENTS']) - np.sum(cbg_prac_grouped['NUMBER_O
F_PATIENTS']),
      ' patients). However while the total number of practices in London is also greater
, the mean number of patients ',
      'per practice is larger in Cambridge (', round(np.mean(cbg_prac_grouped['NUMBER_OF
_PATIENTS'])), ') than London (',
      round(np.mean(ldn_prac_grouped['NUMBER_OF_PATIENTS'])), '), although the differenc
e is not significant:'
      , sep = '')

cbg_mean, var, std = stats.bayes_mvs(cbg_prac_grouped['NUMBER_OF_PATIENTS'], alpha=0.95)
ldn_mean, var, std = stats.bayes_mvs(ldn_prac_grouped['NUMBER_OF_PATIENTS'], alpha=0.95)
print('Cambridge:', cbg_mean)
print('London:', ldn_mean)
```

```
As would be expected the registered population of London is substantially larger than
Cambridge (by 5648283 patients). However while the total number of practices in London
is also greater, the mean number of patients per practice is larger in Cambridge (1005
1) than London (7832), although the difference is not significant:
Cambridge: Mean(statistic=10050.935483870968, minmax=(7748.8816727869225, 12352.989294
955012))
London: Mean(statistic=7831.618922470434, minmax=(7473.6377007316405, 8189.60014420922
7))
```

In [30]:
```
# scrip differences
print('Patients in Cambridge were, on average, prescribed ',
      'more items at a higher average cost. In Cambridge the average prescriptions per p
atients was ',
      round(np.sum(cbg['ITEMS']) / np.sum(cbg_prac_grouped['NUMBER_OF_PATIENTS']), 2), '
items at an cost per person of £',
      round(np.sum(cbg['ACTCOST']) / np.sum(cbg_prac_grouped['NUMBER_OF_PATIENTS']),2),
' while the average total cost of each item was £',
      round(np.mean(cbg['TOTALCOST']),2), ' with an average cost per item of £', round(n
p.mean(cbg['ACTCOST']),2),
      '. This compares to London where the average prescriptions per patient was ',
      round(np.sum(ldn['ITEMS']) / np.sum(ldn_prac_grouped['NUMBER_OF_PATIENTS']), 2), '
items at an cost per person of £',
      round(np.sum(ldn['ACTCOST']) / np.sum(ldn_prac_grouped['NUMBER_OF_PATIENTS']),2),
' while the average total cost of each item was £',
      round(np.mean(ldn['TOTALCOST']),2), ' with an average cost per item of £', round(n
p.mean(ldn['ACTCOST']),2),
      sep = '')
```

```
Patients in Cambridge were, on average, prescribed more items at a higher average cost
. In Cambridge the average prescriptions per patients was 1.11 items at an cost per pe
rson of £7.8 while the average total cost of each item was £1684.34 with an average co
st per item of £60.73. This compares to London where the average prescriptions per pat
ient was 1.0 items at an cost per person of £7.34 while the average total cost of each
item was £1268.13 with an average cost per item of £54.26
```

In [31]:

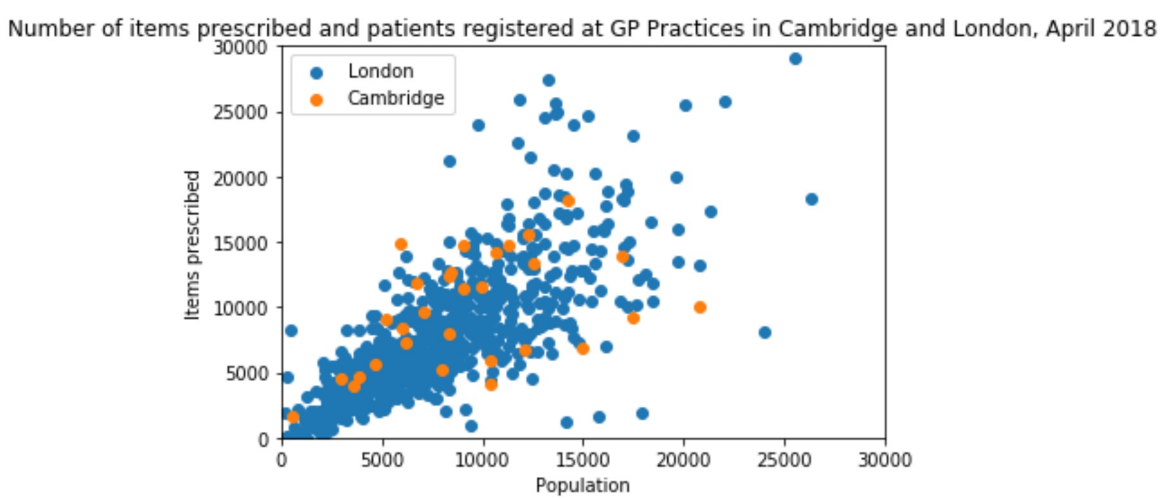
```
# scrips per practice
print('The general trend of bigger practices prescribing more is true in both London and Cambridge. The average prescriptions per ',
      'practice (as opposed to the overall numbers for London and Cambridge) suggests that while the overall prescription rate ',
      'in Cambridge is lower practices as a whole tend to prescribe significantly more in Cambridge than London:',
      sep = '')

cbg_av = cbg_prac_grouped.ITEMS / cbg_prac_grouped.NUMBER_OF_PATIENTS
ldn_av = ldn_prac_grouped.ITEMS / ldn_prac_grouped.NUMBER_OF_PATIENTS

cbg_mean, var, std = stats.bayes_mvs(cbg_av, alpha=0.95)
ldn_mean, var, std = stats.bayes_mvs(ldn_av, alpha=0.95)
print('Cambridge:', cbg_mean)
print('London:', ldn_mean)
```

```
The general trend of bigger practices prescribing more is true in both London and Cambridge. The average prescriptions per practice (as opposed to the overall numbers for London and Cambridge) suggests that while the overall prescription rate in Cambridge is lower practices as a whole tend to prescribe significantly more in Cambridge than London:
Cambridge: Mean(statistic=1.230451977333548, minmax=(1.0255720229140688, 1.4353319317530273))
London: Mean(statistic=1.0752174453282206, minmax=(1.0049231568776256, 1.1455117337788157))
```

In [32]:

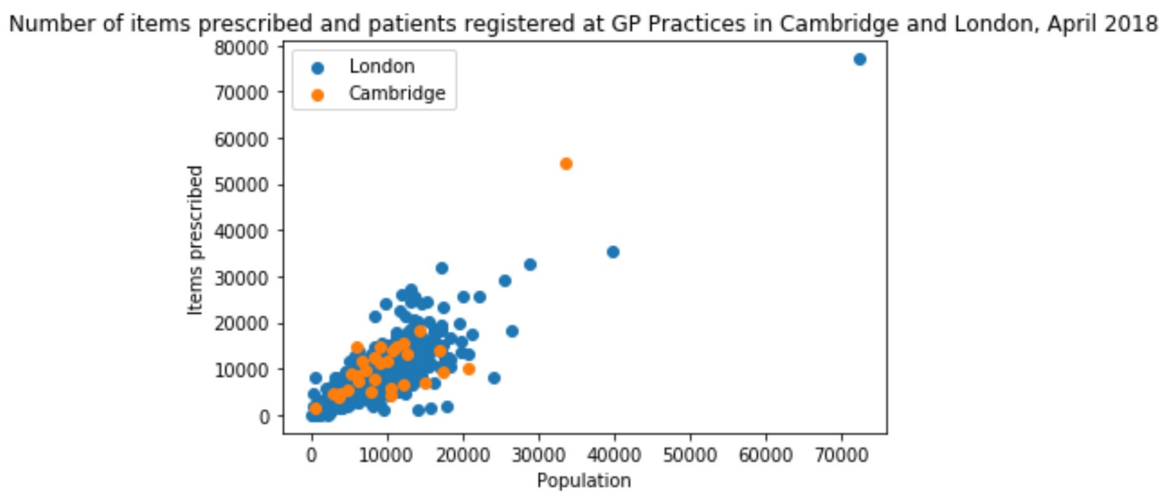
```
# scatter pop vs scrips
_ = plt.scatter(ldn_prac_grouped['NUMBER_OF_PATIENTS'], ldn_prac_grouped['ITEMS'], label = 'London')
_ = plt.scatter(cbg_prac_grouped['NUMBER_OF_PATIENTS'], cbg_prac_grouped['ITEMS'], label = 'Cambridge')
_ = plt.title('Number of items prescribed and patients registered at GP Practices in Cambridge and London, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Items prescribed')
_ = plt.xlim(0,30000)
_ = plt.ylim(0,30000)
_ = plt.legend(loc='upper left')
print('note some extreme values are not shown to highlight the dispersion')
```

```
note some extreme values are not shown to highlight the dispersion
```



In summary London has more patients and practices than Cambridge but the average number of patients per practice in Cambridge is substantially higher. The average number of prescriptions per patient is higher in Cambridge than London, as is the average cost per drug cost per person. This may reflect the different needs of the population in Cambridge. It may also be due to outliers - for example one practice (shown in the graph below) has a higher than expected prescribing rate. However, while there are differences, most Cambridge practices fall within the range of London practices.

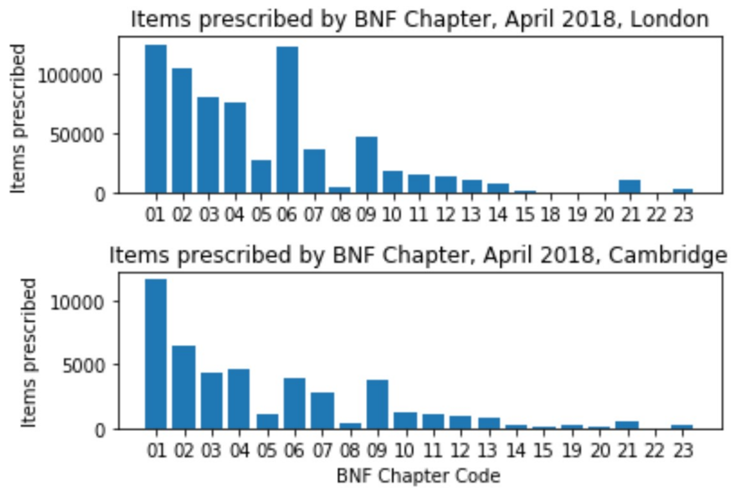
In [33]:
```
# scatter pop vs scrips
_ = plt.scatter(ldn_prac_grouped['NUMBER_OF_PATIENTS'], ldn_prac_grouped['ITEMS'], label
= 'London')
_ = plt.scatter(cbg_prac_grouped['NUMBER_OF_PATIENTS'], cbg_prac_grouped['ITEMS'], label
= 'Cambridge')
_ = plt.title('Number of items prescribed and patients registered at GP Practices in Cam
bridge and London, April 2018')
_ = plt.xlabel('Population')
_ = plt.ylabel('Items prescribed')
_ = plt.legend(loc='upper left')
```



In terms of what is being prescribed there are large differences between London and Cambridge, for example drugs in BNF Chapter 06 makes up a much lower proportion of drugs prescribed in Cambridge than London

In [34]:
```
ldn_bnf_grouped.loc[:,'BNFCHAP'] = ldn_bnf_grouped['BNFCODE'].astype(str).str[0:2]
cbg_bnf_grouped.loc[:,'BNFCHAP'] = cbg_bnf_grouped['BNFCODE'].astype(str).str[0:2]
```

```
In [35]: fig = plt.figure(1)
plt1 = fig.add_subplot(211)
plt1 = plt.bar(ldn_bnf_grouped.BNFCHAP,ldn_bnf_grouped.ITEMS)
plt1 = plt.title('Items prescribed by BNF Chapter, April 2018, London')
plt1 = plt.ylabel('Items prescribed')

plt2 = fig.add_subplot(212)
plt2 = plt.bar(cbg_bnf_grouped.BNFCHAP,cbg_bnf_grouped.ITEMS)
plt2 = plt.title('Items prescribed by BNF Chapter, April 2018, Cambridge')
plt2 = plt.ylabel('Items prescribed')
plt1 = plt.xlabel('BNF Chapter Code')

fig.subplots_adjust(hspace=0.5,wspace=0.5)
plt.show()
```



Prescriptions for cardiovascular and antidepressant drugs

The first two characters of the BNF code are the chapter (chapter 2 starts with 02) and the next two the section (chapter 4 section 3 starts with 0403).

This can be accomplished using regex or pandasql. Code for both is presented and sql used as the package was listed as approved but better performance was obtained with regex.

Cardiovascular

```
In [36]: cvd_sql = """SELECT * FROM scrips_raw WHERE BNFCODE LIKE '02%'"""
scrips_cvd = pysqldf(cvd_sql)

# Select using regex (had better performance)
# scrips_cvd = scrips_raw[scrips_raw.BNFCODE.str.contains("^02", regex=True)]
```

In [37]:
```
print('There were ', np.sum(scrips_cvd.ITEMS), ' items prescribed by primary care establ
ishments (including non-GP practices) for ',
      'cardiovascular drugs (defined as those in chapter 02 of the BNF). A total of ', s
crips_cvd.CODE.nunique(), ' providers prescribed ',
      scrips_cvd.BNFCODE.nunique(), ' different drugs during April 2018 in England. The
total cost of cvd prescriptions was £',
      round(np.sum(scrips_cvd.TOTALCOST),2),
      sep = '')
```

```
There were 26449832 items prescribed by primary care establishments (including non-GP
practices) for cardiovascular drugs (defined as those in chapter 02 of the BNF). A tot
al of 8259 providers prescribed 1230 different drugs during April 2018 in England. The
total cost of cvd prescriptions was £5448656348.37
```

In [38]:
```
# group prescribing data by BNF
scrips_cvd_bnf_grouped = scrips_cvd.groupby(['BNFCODE', 'BNFNAME']).sum()
```

In [39]:
```
print('The 10 most prescribed drugs were:')
scrips_cvd_bnf_grouped.sort_values('ITEMS', axis=0, ascending=False, kind='quicksort', n
a_position='last').head(10)
```

```
The 10 most prescribed drugs were:
```

Out[39]:

		ITEMS	NIC	ACTCOST	QUANTITY	PERIOD	T
BNFCODE	**BNFNAME**						
0206020A0AAAAAA	**Amlodipine_Tab 5mg**	1439739	2541471.95	2392086.43	49595851	1529674320	7.
0212000B0AAABAB	**Atorvastatin_Tab 20mg**	1431023	1280629.42	1223368.38	48157431	1494560424	3.
0209000A0AAABAB	**Aspirin Disper_Tab 75mg**	1415301	515494.96	521502.98	43752171	1491129756	1.
0212000Y0AAADAD	**Simvastatin_Tab 40mg**	1100977	839258.34	808711.85	35932841	1475389044	1.
0212000B0AAACAC	**Atorvastatin_Tab 40mg**	965700	1077775.46	1031034.31	30078354	1488909912	2.
0206020A0AAABAB	**Amlodipine_Tab 10mg**	926675	1796639.47	1690146.43	31711013	1495973052	3.
0205051R0AAADAD	**Ramipril_Cap 10mg**	832680	988669.23	935280.30	29419334	1480232340	1.
0202010B0AAABAB	**Bendroflumethiazide_Tab 2.5mg**	813364	292085.70	288587.92	29181592	1475590848	5.
0209000C0AAAAAA	**Clopidogrel_Tab 75mg**	784229	1014555.17	973762.42	21544252	1488102696	1.
0212000Y0AAABAB	**Simvastatin_Tab 20mg**	763827	404395.65	396569.62	24991556	1471554768	6.

In [40]:
```
print('While the 10 drugs where the most money was spent were:')
scrips_cvd_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=False, kind='quicksort
', na_position='last').head(10)
```

While the 10 drugs where the most money was spent were:

Out[40]:

		ITEMS	NIC	ACTCOST	QUANTITY	PERIOD	TOTAL
BNFCODE	**BNFNAME**						
0206020A0AAAAAA	**Amlodipine_Tab 5mg**	1439739	2541471.95	2392086.43	49595851	1529674320	7.09942
0208020Y0AAACAC	**Rivaroxaban_Tab 20mg**	228645	11954784.60	11094375.29	6641547	1455814056	6.65272
0208020Z0AAABAB	**Apixaban_Tab 5mg**	219971	11389346.70	10569680.46	11988786	1457630292	6.33411
0212000B0AAABAB	**Atorvastatin_Tab 20mg**	1431023	1280629.42	1223368.38	48157431	1494560424	3.76340
0206020A0AAABAB	**Amlodipine_Tab 10mg**	926675	1796639.47	1690146.43	31711013	1495973052	3.22805
0212000B0AAACAC	**Atorvastatin_Tab 40mg**	965700	1077775.46	1031034.31	30078354	1488909912	2.13642
0212000Y0AAADAD	**Simvastatin_Tab 40mg**	1100977	839258.34	808711.85	35932841	1475389044	1.99983
0205051R0AAADAD	**Ramipril_Cap 10mg**	832680	988669.23	935280.30	29419334	1480232340	1.98728
0208020Z0AAAAAA	**Apixaban_Tab 2.5mg**	127975	5554284.25	5161392.03	5846615	1402941408	1.77378
0209000C0AAAAAA	**Clopidogrel_Tab 75mg**	784229	1014555.17	973762.42	21544252	1488102696	1.65532

Antidepressant

In [41]:
```
dep_sql = """SELECT * FROM scrips_raw WHERE BNFCODE LIKE '0403%'"""
scrips_dep = pysqldf(dep_sql)

# Select using regex (had better performance)
# scrips_dep = scrips_raw[scrips_raw.BNFCODE.str.contains("^0403", regex=True)]
```

In [42]:
```
print('There were ', np.sum(scrips_dep.ITEMS), ' items prescribed by primary care establ
ishments (including non-GP practices) for ',
      'antidepressant drugs (defined as those in chapter 0403 of the BNF). A total of ',
scrips_dep.CODE.nunique(), ' providers prescribed ',
      scrips_dep.BNFCODE.nunique(), ' different drugs during April 2018 in England. The
total cost of antidepressant prescriptions was £',
      round(np.sum(scrips_dep.TOTALCOST),2),
      sep = '')
```

```
There were 5715873 items prescribed by primary care establishments (including non-GP p
ractices) for antidepressant drugs (defined as those in chapter 0403 of the BNF). A to
tal of 8148 providers prescribed 201 different drugs during April 2018 in England. The
total cost of antidepressant prescriptions was £925174735.92
```

In [43]:
```
# group prescribing data by BNF
scrips_dep_bnf_grouped = scrips_dep.groupby(['BNFCODE', 'BNFNAME']).sum()
```

In [44]:
```
print('The 10 most presbribed drugs were:')
scrips_dep_bnf_grouped.sort_values('ITEMS', axis=0, ascending=False, kind='quicksort', n
a_position='last').head(10)
```

```
The 10 most presbribed drugs were:
```

Out[44]:

		ITEMS	NIC	ACTCOST	QUANTITY	PERIOD	TOTALCOS
BNFCODE	**BNFNAME**						
0403010B0AAAGAG	**Amitriptyline HCl_Tab 10mg**	701543	1261322.48	1187746.02	36385621	1546020444	2.001345e+(
0403030D0AAAAAA	**Citalopram Hydrob_Tab 20mg**	671305	1883443.45	1763522.77	22524573	1513731804	2.663320e+(
0403030Q0AAAAAA	**Sertraline HCl_Tab 50mg**	634023	486330.38	469886.38	20122418	1542993384	6.873814e+(
0403030Q0AAABAB	**Sertraline HCl_Tab 100mg**	531490	591344.34	565358.74	17765869	1521400356	7.061664e+(
0403030E0AAAAAA	**Fluoxetine HCl_Cap 20mg**	509330	444712.64	443740.91	22945968	1510301136	5.497144e+(
0403030D0AAABAB	**Citalopram Hydrob_Tab 10mg**	331855	578064.68	544841.77	10710078	1497183876	4.160111e+(
0403010B0AAAHAH	**Amitriptyline HCl_Tab 25mg**	247551	297480.32	283395.27	11237854	1471958376	1.801838e+(
0403040X0AAANAN	**Mirtazapine_Tab 15mg**	233935	489792.07	464020.56	6012211	1491936972	2.605892e+(
0403040X0AAAAAA	**Mirtazapine_Tab 30mg**	223553	197583.56	193214.68	5309652	1485479244	1.031282e+(
0403040X0AAAPAP	**Mirtazapine_Tab 45mg**	195736	351046.52	335742.48	4234235	1456621272	1.725996e+(

In [45]:
```
print('While the 10 drugs where the most money was spent were:')
scrips_dep_bnf_grouped.sort_values('TOTALCOST', axis=0, ascending=False, kind='quicksort
', na_position='last').head(10)
```

While the 10 drugs where the most money was spent were:

Out[45]:

		ITEMS	NIC	ACTCOST	QUANTITY	PERIOD	TOTALCOS
BNFCODE	**BNFNAME**						
0403030D0AAAAAA	**Citalopram Hydrob_Tab 20mg**	671305	1883443.45	1763522.77	22524573	1513731804	2.663320e+(
0403010B0AAAGAG	**Amitriptyline HCl_Tab 10mg**	701543	1261322.48	1187746.02	36385621	1546020444	2.001345e+(
0403030Q0AAABAB	**Sertraline HCl_Tab 100mg**	531490	591344.34	565358.74	17765869	1521400356	7.061664e+(
0403030Q0AAAAAA	**Sertraline HCl_Tab 50mg**	634023	486330.38	469886.38	20122418	1542993384	6.873814e+(
0403030E0AAAAAA	**Fluoxetine HCl_Cap 20mg**	509330	444712.64	443740.91	22945968	1510301136	5.497144e+(
0403040Y0AAABAB	**Duloxetine HCl_Cap G/R 60mg**	119559	1514857.85	1409795.66	3661872	1404152232	4.897224e+(
0403030D0AAABAB	**Citalopram Hydrob_Tab 10mg**	331855	578064.68	544841.77	10710078	1497183876	4.160111e+(
0403040X0AAANAN	**Mirtazapine_Tab 15mg**	233935	489792.07	464020.56	6012211	1491936972	2.605892e+(
0403010B0AAAIAI	**Amitriptyline HCl_Tab 50mg**	154237	641513.88	600310.01	6221348	1437046284	2.322865e+(
0403010B0AAAHAH	**Amitriptyline HCl_Tab 25mg**	247551	297480.32	283395.27	11237854	1471958376	1.801838e+(

Prescribing in April 2018

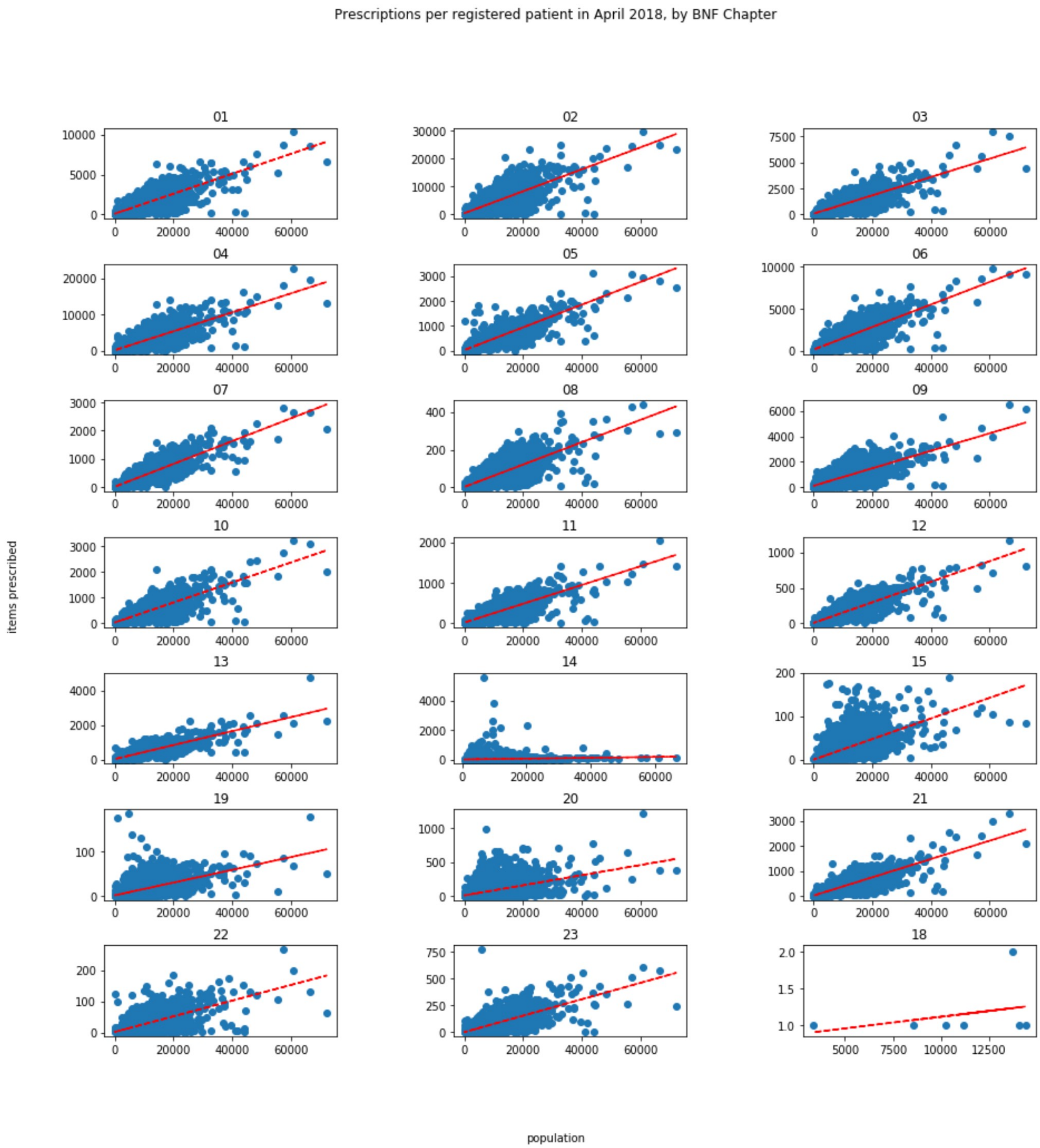
In [46]:
```
#filter prescriptions to practices with pop data only
cols = ['CODE', 'NUMBER_OF_PATIENTS', 'BNFCODE', 'BNFNAME', 'ITEMS', 'ACTCOST', 'QUANTIT
Y', 'TOTALCOST']
scrips_0418 = pd.merge(pop_raw, scrips_raw, on='CODE', how='inner')
scrips_0418 = scrips_0418[cols]
scrips_0418.loc[:,'BNFCHAP'] = scrips_0418['BNFCODE'].astype(str).str[0:2]
```

In [47]:
```
# group prescribing data by practice and BNF Chapter
scrips_0418_grouped = scrips_0418.drop(['ACTCOST'], axis=1).groupby(['CODE', 'NUMBER_OF_
PATIENTS', 'BNFCHAP']).sum().reset_index()
```

In [48]:
```
print('In April 2018, across all practices in England a total of ', np.sum(scrips_0418.I
TEMS), ' were prescribed, made up of ',
      scrips_0418.BNFCODE.nunique(), ' different drugs. We can see below that there is a
strong trend for bigger practices to prescribe more.',
      sep = '')
```

```
In April 2018, across all practices in England a total of 87842876 were prescribed, ma
de up of 22116 different drugs. We can see below that there is a strong trend for bigg
er practices to prescribe more.
```

In [49]:
```
#plots of presciptions by population
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

bnfchaps = scrips_0418_grouped.BNFCHAP.unique()

for chap,num in zip(bnfchaps, range(1,22)):
    df = scrips_0418_grouped[scrips_0418_grouped['BNFCHAP']==chap]
    ax = fig.add_subplot(7,3,num)
    ax.scatter(df.NUMBER_OF_PATIENTS, df.ITEMS)
    ax.set_title(chap)
    z = np.polyfit(df.NUMBER_OF_PATIENTS, df.ITEMS, 1)
    p = np.poly1d(z)
    plt.plot(df.NUMBER_OF_PATIENTS,p(df.NUMBER_OF_PATIENTS),"r--")

fig.text(0.5,0.04, "population", ha="center", va="center")
fig.text(0.05,0.5, "items prescribed", ha="center", va="center", rotation=90)


fig.subplots_adjust(hspace=0.5,wspace=0.5)
fig.suptitle("Prescriptions per registered patient in April 2018, by BNF Chapter")
plt.show()
```

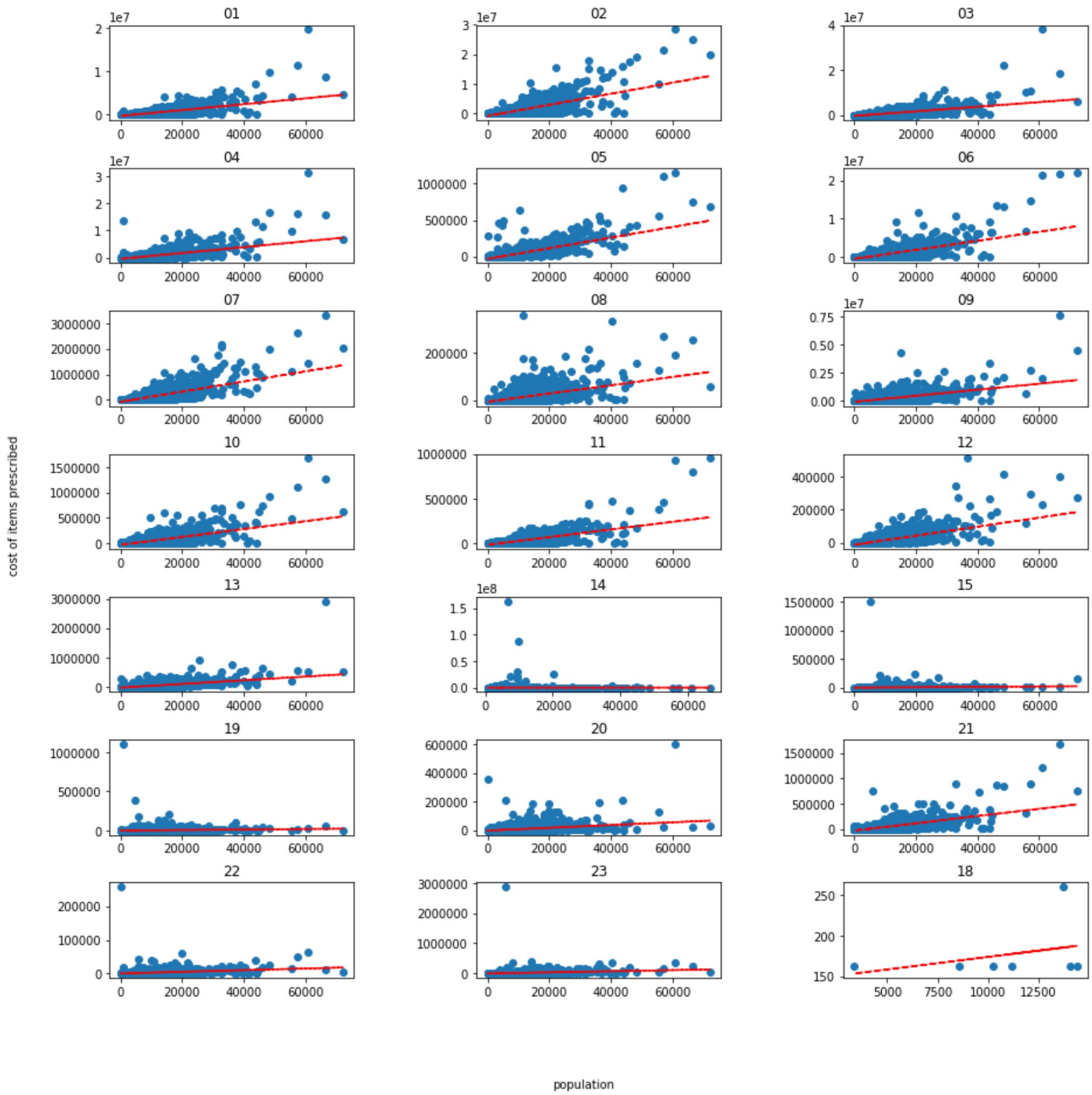


This also translates into cost, though as with quantity above there are large differences between chapters.

In [50]:

```
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

bnfchaps = scrips_0418_grouped.BNFCHAP.unique()

for chap,num in zip(bnfchaps, range(1,22)):
    df = scrips_0418_grouped[scrips_0418_grouped['BNFCHAP']==chap]
    ax = fig.add_subplot(7,3,num)
    ax.scatter(df.NUMBER_OF_PATIENTS, df.TOTALCOST)
    ax.set_title(chap)
    z = np.polyfit(df.NUMBER_OF_PATIENTS, df.TOTALCOST, 1)
    p = np.poly1d(z)
    plt.plot(df.NUMBER_OF_PATIENTS,p(df.NUMBER_OF_PATIENTS),"r--")

fig.text(0.5,0.04, "population", ha="center", va="center")
fig.text(0.05,0.5, "cost of items prescribed", ha="center", va="center", rotation=90)

fig.subplots_adjust(hspace=0.5,wspace=0.5)
fig.suptitle("Cost of items prescribed per registered patient in April 2018, by BNF Chap
ter")
plt.show()
```

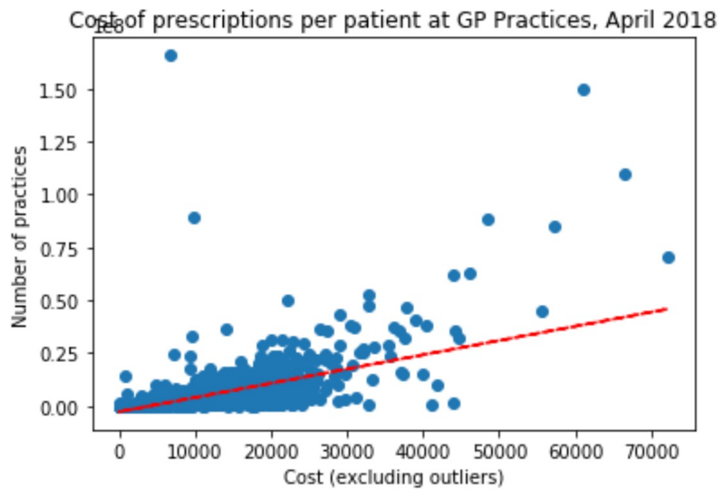
Cost of items prescribed per registered patient in April 2018, by BNF Chapter




```
In [51]: scrips_0418.loc[:,'COSTPP'] = scrips_0418['TOTALCOST'] / scrips_0418['NUMBER_OF_PATIENTS
'']
```

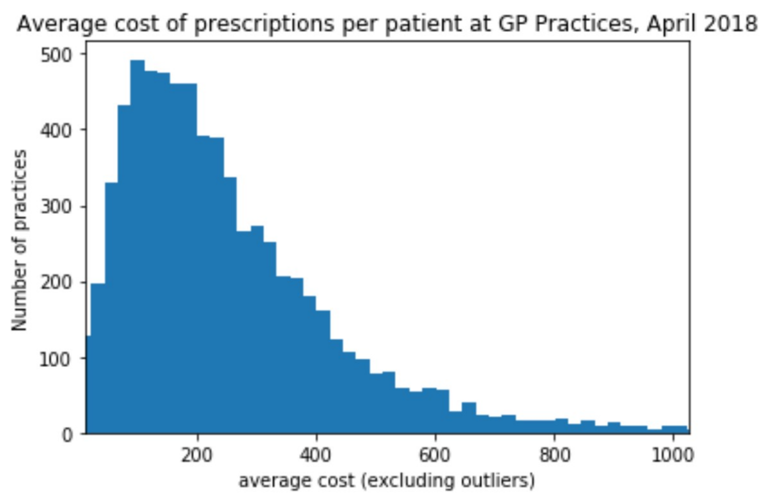
```
In [52]: scrips_0418_prac_grouped = scrips_0418.drop(['ACTCOST'], axis=1).groupby(['CODE', 'NUMBE
R_OF_PATIENTS']).sum().reset_index('NUMBER_OF_PATIENTS')
```

In [53]:
```
#check axis labels
_ = plt.scatter(scrips_0418_prac_grouped.NUMBER_OF_PATIENTS,scrips_0418_prac_grouped.TOTALCOST)
_ = plt.title('Cost of prescriptions per patient at GP Practices, April 2018')
_ = plt.xlabel('Cost (excluding outliers)')
_ = plt.ylabel('Number of practices')
z = np.polyfit(scrips_0418_prac_grouped.NUMBER_OF_PATIENTS, scrips_0418_prac_grouped.TOTALCOST, 1)
p = np.poly1d(z)
plt.plot(scrips_0418_prac_grouped.NUMBER_OF_PATIENTS,p(scrips_0418_prac_grouped.NUMBER_OF_PATIENTS),"r--")
plt.show()
```



Looking overall the cost of prescriptions increases with larger practice size, although there is variation.

In [54]:
```
min_x = scrips_0418_prac_grouped.COSTPP.quantile(.01)
max_x = scrips_0418_prac_grouped.COSTPP.quantile(.99)
_ = plt.hist(scrips_0418_prac_grouped.COSTPP, bins = 'auto')
_ = plt.xlim(min_x, max_x)
_ = plt.title('Average cost of prescriptions per patient at GP Practices, April 2018')
_ = plt.xlabel('average cost (excluding outliers)')
_ = plt.ylabel('Number of practices')
```



In [55]:
```
scrips_0418_prac_grouped.COSTPP.describe()
```

Out[55]:
```
count     7191.000000
mean       280.180256
std        800.646710
min          0.001724
25%        122.804409
50%        208.967847
75%        337.701855
max      44745.505000
Name: COSTPP, dtype: float64
```

In [56]:
```
scrips_0418_prac_grouped.sort_values('COSTPP',ascending=False).head()
```

Out[56]:

	NUMBER_OF_PATIENTS	ITEMS	QUANTITY	TOTALCOST	COSTPP
CODE					
Y02045	8	1054	136990	3.579640e+05	44745.505000
Y02873	3	1537	119971	9.640291e+04	32134.303333
P82010	6694	19125	1000424	1.657064e+08	24754.467831
A81630	752	5179	1314051	1.390687e+07	18493.175093
Y01924	106	1714	2977949	1.712883e+06	16159.271415

For the most part practices spent less than £300 per patient on prescriptions in April 2018, although there were significant outliers which look to have special circumstances (e.g. low number of patients, serving non-typical patient groups).

Conclusions

While larger practices prescribe more, as would be expected, there is substantial variation between practices both in terms of prescriptions per patient and what is prescribed. Where this variation is unwarranted action could be taken better target prescribing on patients who require it most and encourage the use of cheaper medications. Action could also be taken encourage the use of standard medications to reduce the variety of medications and pack sizes that are stocked; while this may increase cost in terms of prescribing more than is needed it may have savings for the supply line through stocking less variety.

Part B: WHO Mortality

Setup

In [57]:
```
# import packages
# import numpy as np
# import matplotlib.pyplot as plt
# from scipy import stats
# import pandas as pd
```

In [58]:
```
# import country data
country_codes_raw = pd.read_csv('who/country_codes.csv')
country_codes = country_codes_raw.loc[(country_codes_raw['name'] == 'Iceland') |
                                      (country_codes_raw['name'] == 'Italy') |
                                      (country_codes_raw['name'] == 'New Zealand') |
                                      (country_codes_raw['name'] == 'Australia')]
```

In [59]:
```
# import pop data and filter to countries of interest
colnames = ['country', 'admin1', 'subdiv1', 'year', 'sex', 'format', 'all_age',
           '0-0', '1-1', '2-2', '3-3', '4-4', '5-9', '10-14', '15-19', '20-24',
           '25-29', '30-34', '35-39', '40-44', '45-49', '50-54', '55-59', '60-64',
           '65-69', '70-74', '75-79', '80-84', '85-89', '90-94', '95+', 'unspecified',
           'livebirths']
pop_raw = pd.read_csv('who/pop.csv', header = 0, names = colnames)
pop = pd.merge(pop_raw, country_codes, on='country', how='inner')
```

In [60]:
```
# import and combine deaths data and filter to countries of interest
colnames = ['country', 'admin1', 'subdiv1', 'year', 'list', 'cause', 'sex', 'format', 'i
nfant_format',
            'all_age', '0-0', '1-1', '2-2', '3-3', '4-4', '5-9', '10-14', '15-19',
            '20-24', '25-29', '30-34', '35-39', '40-44', '45-49', '50-54', '55-59',
            '60-64', '65-69', '70-74', '75-79', '80-84', '85-89', '90-94', '95+',
            'unspecified', 'infant_0', 'infant_1-6', 'infant_1-27', 'infant_28-364']
deaths1 = pd.read_csv('who/Morticd10_part1.csv', header = 0, names = colnames,
                      dtype = {'admin1': np.str, 'subdiv1':np.str, 'list': np.str})
deaths2 = pd.read_csv('who/Morticd10_part2.csv', header = 0, names = colnames,
                      dtype = {'admin1': np.str, 'subdiv1':np.str, 'list': np.str})
deaths_comb = deaths1.append(deaths2)
deaths = pd.merge(deaths_comb, country_codes, on='country', how='inner')
```

Population & deaths in Iceland, Italy and New Zealand

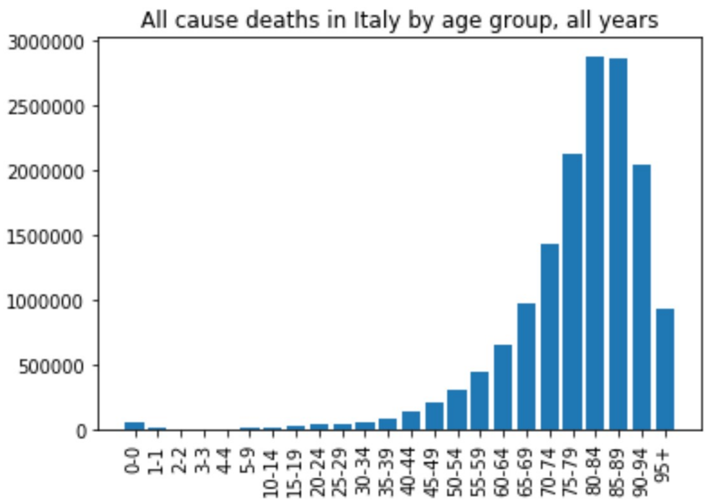
In [61]:
```
# pop and deaths in 2010 for each country then chi2
ice_pop = np.sum(pop.all_age[(pop.name == 'Iceland') & (pop.year == 2010)])
ice_dea = np.sum(deaths.all_age[(deaths.name == 'Iceland') & (deaths.year == 2010)])
ita_pop = np.sum(pop.all_age[(pop.name == 'Italy') & (pop.year == 2010)])
ita_dea = np.sum(deaths.all_age[(deaths.name == 'Italy') & (deaths.year == 2010)])
new_pop = np.sum(pop.all_age[(pop.name == 'New Zealand') & (pop.year == 2010)])
new_dea = np.sum(deaths.all_age[(deaths.name == 'New Zealand') & (deaths.year == 2010)])

print('The population of Iceland in 2010 was', ice_pop, 'while the number of deaths was'
, ice_dea,
      'meaning that the % of the population that died was', round(ice_dea / ice_pop * 10
0, 2) , '%')
print('The population of Italy in 2010 was', ita_pop, 'while the number of deaths was',
ita_dea,
      'meaning that the % of the population that died was', round(ita_dea / ita_pop * 10
0, 2) , '%')
print('The population of New Zealand in 2010 was', new_pop, 'while the number of deaths
was', new_dea,
      'meaning that the % of the population that died was', round(new_dea / new_pop * 10
0, 2) , '%')

chi2 = [[ice_pop, ice_dea], [ita_pop, ita_dea], [new_pop, new_dea]]
chi2, p, dof, ex = stats.chi2_contingency(chi2)
print('chi2:', round(chi2, 2), 'p:', p)
```

```
The population of Iceland in 2010 was 318041.0 while the number of deaths was 4038 mea
ning that the % of the population that died was 1.27 %
The population of Italy in 2010 was 60483386.0 while the number of deaths was 1169230
meaning that the % of the population that died was 1.93 %
The population of New Zealand in 2010 was 4367360.0 while the number of deaths was 572
98 meaning that the % of the population that died was 1.31 %
chi2: 8851.6 p: 0.0
```

The overall rate of deaths across the three countries is significantly different but the differences are quite small.

Distribution of deaths in Italy

In [62]:
```
# chart of all cause deaths in Italy across all years by age
cols = ['0-0', '1-1', '2-2', '3-3', '4-4', '5-9', '10-14', '15-19',
            '20-24', '25-29', '30-34', '35-39', '40-44', '45-49', '50-54', '55-59',
            '60-64', '65-69', '70-74', '75-79', '80-84', '85-89', '90-94', '95+',]

deaths_italy_sum = deaths[deaths.name == 'Italy'].groupby(['name']).sum().transpose().re
set_index()
deaths_italy_pivot = deaths_italy_sum[deaths_italy_sum['index'].isin(cols)]

_ = plt.bar(deaths_italy_pivot['index'], deaths_italy_pivot['Italy'])
_ = plt.xticks(rotation=90)
_ = plt.title('All cause deaths in Italy by age group, all years')
```



As would be expected the number of deaths from neoplasms in Italy across all years increases with age, failing off from 85+ as the total population declines and there are fewer people to die of neoplasms

Deaths from neoplasms in Italy

In [63]:
```
# total number of deaths in Italy
ita_deaths_total = deaths[deaths.name == 'Italy'].all_age.sum()
```

In [64]:
```
#filter to italy neoplasms
deaths_italy_group_neop = deaths[(deaths.cause >= 'C000') & (deaths.cause <= 'D489') & (
deaths.name == 'Italy')]
#group, select columns and calcaute props
deaths_italy_group_neop = deaths_italy_group_neop[['cause','all_age']].groupby(['cause']
).sum().reset_index()
deaths_italy_group_neop.loc[:,'prop of neoplasm deaths'] = deaths_italy_group_neop.all_a
ge / deaths_italy_group_neop.all_age.sum() * 100
deaths_italy_group_neop.loc[:,'prop of all deaths'] = deaths_italy_group_neop.all_age /
ita_deaths_total * 100
```

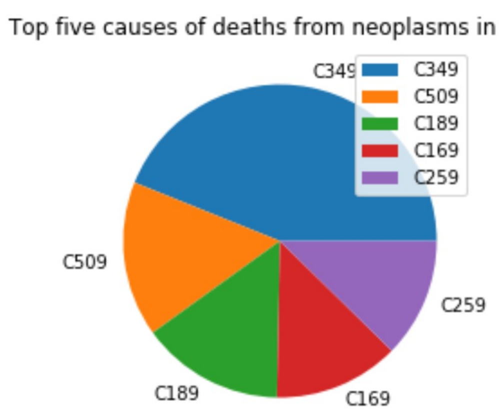
In [65]:
```
#most common five neoplasm causes
print('The five most common causes of deaths from neoplams (ICD-10 C00-D48) in Italy wer
e:')
deaths_italy_group_neop5 = deaths_italy_group_neop.sort_values('prop of neoplasm deaths'
, ascending=False).head()
deaths_italy_group_neop5
```

```
The five most common causes of deaths from neoplams (ICD-10 C00-D48) in Italy were:
```

Out[65]:

	cause	all_age	prop of neoplasm deaths	prop of all deaths
143	C349	426451	18.964664	2.790770
227	C509	155895	6.932792	1.020204
92	C189	143188	6.367701	0.937047
76	C169	125679	5.589059	0.822465
118	C259	120070	5.339622	0.785759

In [66]:
```
# pie chart of 5 most common
_ = plt.pie(deaths_italy_group_neop5.all_age, labels=deaths_italy_group_neop5.cause)
_ = plt.legend(deaths_italy_group_neop5.cause)
_ = plt.title('Top five causes of deaths from neoplasms in Italy')
_ = plt.show()
```



The five most common causes of death from neoplasm explain a small percentage of the deaths from all causes, though around 40% of the deaths from all types of neoplasms which overall accounted for a large segment of deaths in Italy.

In [67]:
```
print(round(deaths_italy_group_neop['all_age'].sum() / ita_deaths_total * 100,2), '% of
all deaths were caused by neoplasms')
```

```
14.72 % of all deaths were caused by neoplasms
```

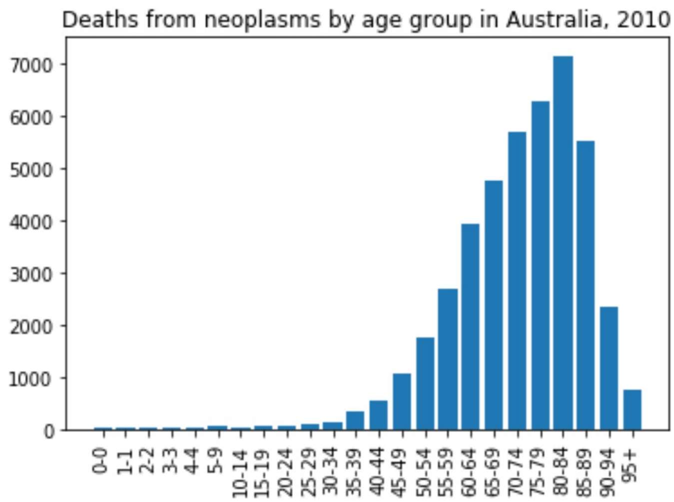
Deaths from neosplasms in Australia during 2010 by age

In [68]:
```
# Australian neoplasm deaths by age
deaths_aus_sum = deaths[(deaths.name == 'Australia') & (deaths.year == 2010) & (deaths.c
ause >= 'C000') & (deaths.cause <= 'D489')].groupby(['name']).sum().transpose().reset_in
dex()
deaths_aus_pivot = deaths_aus_sum[deaths_aus_sum['index'].isin(cols)]
```

In [69]:
```
# Australian population by age
pop_aus_sum = pop[(pop.name == 'Australia') & (pop.year == 2010)].groupby(['name']).sum(
).transpose().reset_index()
pop_aus_pivot = pop_aus_sum[pop_aus_sum['index'].isin(cols)]
```

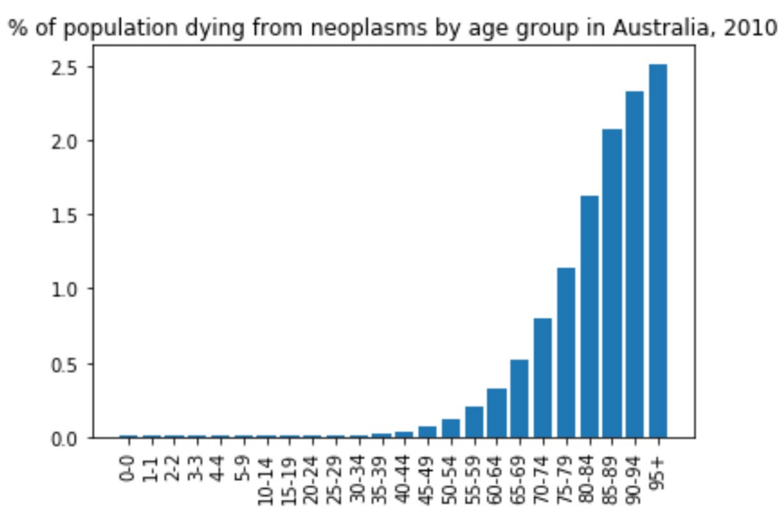
In [70]:
```
# merge deaths and pop and add prop
aus_neop = pd.merge(pop_aus_pivot, deaths_aus_pivot, on='index', suffixes=('_pop', '_dea
ths'))
aus_neop.loc[:,'prop'] = aus_neop.Australia_deaths / aus_neop.Australia_pop * 100
```

In [71]:
```
# hist of number of deaths
_ = plt.bar(aus_neop['index'], aus_neop.Australia_deaths)
_ = plt.xticks(rotation=90)
_ = plt.title('Deaths from neoplasms by age group in Australia, 2010')
_ = plt.show()
```



There are substantially higher number of deaths from neoplasms in older people in Australia during 2010 though this decreases in the 85+ groups because of the smaller population as shown in the following graph

In [72]:
```
# hist of prop of deaths
_ = plt.bar(aus_neop['index'], aus_neop.prop)
_ = plt.xticks(rotation=90)
_ = plt.title('% of population dying from neoplasms by age group in Australia, 2010')
_ = plt.show()
```



While the raw number of deaths is lower in the 85+ age groups than the 70+ age groups there are substantially higher proportions of deaths from neoplasms in older people in Australia during 2010

Variation in deaths from neoplasms in Australia and Italy during 2010

In [73]:
```
# filter to neoplasm deaths in Aus/Ita in 2010
deaths_neop = deaths[((deaths.name == 'Australia') | (deaths.name == 'Italy')) & (deaths
.year == 2010) & (deaths.cause >= 'C000') & (deaths.cause <= 'D489')]

#filter columns and melt deaths
cols.append('name'), cols.append('sex'), cols.append('cause')
deaths_neop = deaths_neop[[c for c in deaths_neop.columns if c in cols]].groupby(['name'
, 'cause', 'sex']).sum().reset_index()
deaths_neop = pd.melt(deaths_neop, id_vars = ['name', 'sex', 'cause'], var_name = 'age_g
roup', value_name = 'deaths')
```

In [74]:
```
# filter and melt pop data
pop_ausita = pop[((pop.name == 'Australia') | (pop.name == 'Italy')) & (pop.year == 2010
)]
pop_ausita = pop_ausita[[c for c in pop_ausita.columns if c in cols]].groupby(['name', '
sex']).sum().reset_index()
pop_ausita = pd.melt(pop_ausita, id_vars = ['name', 'sex'], var_name = 'age_group', valu
e_name = 'population')
```

In [75]:
```
# compare total neoplasm deaths
totals = pd.DataFrame({'pop':np.array(pop_ausita['population'].groupby(pop_ausita.name).
sum()),
                     'total_deaths':np.array(deaths_neop['deaths'].groupby(deaths_neop.n
ame).sum()),
                      'name':['Australia','Italy']})
totals.loc[:, 'prop'] = totals['total_deaths'] / totals['pop'] * 100
print(totals.set_index('name'))
chi2, p, dof, ex = stats.chi2_contingency(totals.iloc[:,0:2])
print('chi2:', chi2, 'p:', p)
```

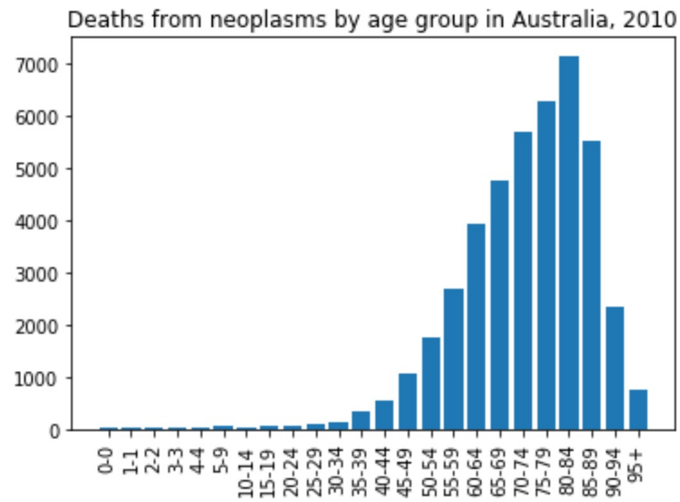
```
                  pop  total_deaths     prop
name
Australia  22297515.0       43313.0  0.19425
Italy      60483396.0      175045.0  0.28941
chi2: 5580.16950002019 p: 0.0
```

A significantly higher proportion of the population died from neoplasms in Italy during 2010 than Australia ($p<0.05$) suggesting some differences in the makeup of the population, different risk factors or treatments.

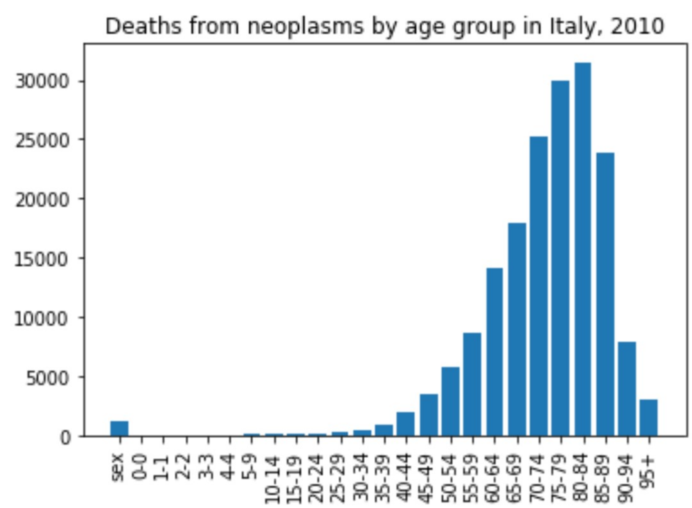
By age

In [76]:
```
# Italy neoplasm deaths by age
deaths_ita_sum = deaths[(deaths.name == 'Italy') & (deaths.year == 2010) & (deaths.cause
>= 'C000') & (deaths.cause <= 'D489')].groupby(['name']).sum().transpose().reset_index()
deaths_ita_pivot = deaths_ita_sum[deaths_aus_sum['index'].isin(cols)]
# Italy population by age
pop_ita_sum = pop[(pop.name == 'Italy') & (pop.year == 2010)].groupby(['name']).sum().tr
anspose().reset_index()
pop_ita_pivot = pop_ita_sum[pop_ita_sum['index'].isin(cols)]
ita_neop = pd.merge(pop_ita_pivot, deaths_ita_pivot, on='index', suffixes=('_pop', '_dea
ths'))
```

In [77]:
```
# Aus hist by age
_ = plt.bar(aus_neop['index'], aus_neop.Australia_deaths)
_ = plt.xticks(rotation=90)
_ = plt.title('Deaths from neoplasms by age group in Australia, 2010')
```



In [78]:
```
# Ita hist by age
_ = plt.bar(ita_neop['index'], ita_neop.Italy_deaths)
_ = plt.xticks(rotation=90)
_ = plt.title('Deaths from neoplasms by age group in Italy, 2010')
_ = plt.show()
```



Despite the differences in the total numbers and proportions dying from neoplasms the proportion by age group is quite similar between the countries suggesting a particular age group is not accounting for the higher overall neoplasm deaths in Italy and it is spread out across age bands.

By sex

In [79]:
```
# group by sex and merge with totals
sex = deaths_neop.loc[:, ['name', 'sex', 'deaths']].groupby(['name', 'sex']).sum().reset
_index()
sex = pd.merge(sex, totals, on=['name'])
# add prop
sex.loc[:, 'prop'] = sex.deaths / sex.total_deaths * 100
sex
```

Out[79]:

	name	sex	deaths	pop	total_deaths	prop
0	Australia	1	24557.0	22297515.0	43313.0	56.696604
1	Australia	2	18756.0	22297515.0	43313.0	43.303396
2	Italy	1	98846.0	60483396.0	175045.0	56.468908
3	Italy	2	76199.0	60483396.0	175045.0	43.531092

In both countries men made up a greater proportion of total neoplasm deaths again suggesting a particular sex is not accounting for the higher deaths in Italy.

By cause

In [107]:
```
# group by cause, merge with totals, add prop
cause = deaths_neop.loc[:, ['name', 'cause', 'deaths']].groupby(['name', 'cause']).sum(
).reset_index()
cause = pd.merge(cause, totals, on='name', how='left')
cause.loc[:, 'prop of total deaths'] = cause.deaths / cause.total_deaths * 100
cause = cause.drop('prop', axis = 1)
```

In [108]:
```
# top 5 for Aus
cause.loc[cause['name'] == 'Australia'].sort_values('prop of total deaths', ascending=F
alse).head()
```

Out[108]:

	name	cause	deaths	pop	total_deaths	prop of total deaths
93	Australia	C349	7989.0	22297515.0	43313.0	18.444809
163	Australia	C61	3236.0	22297515.0	43313.0	7.471198
148	Australia	C509	2865.0	22297515.0	43313.0	6.614642
216	Australia	C809	2783.0	22297515.0	43313.0	6.425323
72	Australia	C259	2367.0	22297515.0	43313.0	5.464872

In [109]:
```
# top 5 for Ita
cause.loc[cause['name'] == 'Italy'].sort_values('prop of total deaths', ascending=False
).head()
```

Out[109]:

	name	cause	deaths	pop	total_deaths	prop of total deaths
457	Italy	C349	33416.0	60483396.0	175045.0	19.089948
520	Italy	C509	12231.0	60483396.0	175045.0	6.987346
408	Italy	C189	11638.0	60483396.0	175045.0	6.648576
432	Italy	C259	9683.0	60483396.0	175045.0	5.531720
393	Italy	C169	9523.0	60483396.0	175045.0	5.440315

Three of the top five neoplasm mortality causes were common across Italy and Australia and they made up a similar % of the total deaths from neoplasms. It is possible that risk, treatment or coding differences explain the differences (e.g. Italy has more neoplasms not otherwise classified so may not code as precisely)