Health Data Reusability Project

This notebook is an informal investigation into the technologies needed to take the data contained in "open data" publications from the UK Department of Health. This will allow researchers to automate computations and respond more speedily to changes.

Even this published product has been subject to much revision and polishing to eliminate simple experiments and code written purely to understand certain aspects of various package's behavior. My bad habit is to delete this code before it is committed to a source control system, but I would recommend that you inculcate better habits than me. And, another thing I don't do enough: commit _[saves sources and checks into dev branch befre continuing].

Some of this is ugly, some of it will doubtless be unnecessary, but it shows at least some of the preliminary work that goes into getting one's thinking straightened out about a particular program or set of programs. I have tried to resist the urge to return and “polish” the code, bit have reserved the right to retrospectively revisit the narrative to augment its educational (and entertainment, for the two go hand in hand) value.

This notebook is also the basis of a 25-minute talk planned for PyData London on 4 August, 2015.

Steve Holden

Acknowledgments

I am grateful to all contributors to the Python developer ecosystem and community.

The authors of and contributors to the [Pandas], [openpyxl], and [Jupyter] projects deserve our undying thanks and continued support for the valuable tools they are creating to make science more developmental and development more scientific. I must also mention Fredrik Lund [the effbot], the forgotten hero who gave us PIL, now re-invigorated as Pillow. It is still a delight to produce simple images with simple code.

I am grateful to the UK government, through its agents, for making the data available, and would be delighted if this work allows them too to use the data more flexibly and to better purpose.

In [1]:

```
import openpyxl as xl
```

Note that this software cannot read ".xls" files. `wb = xl.load_workbook("data/gpearnextime.xls")` raises an exception, so I toook the quick route and converted it to a ".xlsx" file with Excel before further processing. Extending its capabilities would be useful, but would arguably be prolonging the life of a file format the UK government should no longer be permitting.

One might investigate the older `xlrd` module, which can read ".xls" files (though sadly there appears to be no easy way to write them out as ".xslx" files which I had hoped `xlwt` might have provided). It may also be possible to simply use `openpyxl` for `.xlsx` files or `xlrd` for `.xls` files, but that would require feasibility studies and engineering work (though perhaps not too much).

Since this is an experimental project, I punted on this issue and manually performed a function that does not seem amenable to early automation. So sue me.

In [2]:

```
wb = xl.load_workbook("data/gpearnextime.xlsx")
```

```
/Users/sholden/Projects/HealthData/hd.env/lib/python3.4/site-pa
ckages/openpyxl/workbook/names/named_range.py:121: UserWarning:
Discarded range with reserved name
  warnings.warn("Discarded range with reserved name")
```

Ironically, during development I discovered that `Pandas` can (naturally) read Excel spreadheets, and presumed I no longer needed `openpyxl`. I later realized that because Pandas returns a `dict` of spreadsheets I needed `openpyxl` to determine the ordering of the sheets

In [3]:

```
wb.sheetnames
```

Out[3]:

```
['Contents',
 'Definitions and Changes',
 '1a. GPMS Cash Terms ',
 '1b. GPMS Real Terms',
 'Real terms working - HIDE',
 '1c. GMS',
 '1d. PMS',
 '2a. GPMS Expenses',
 '2b. GMS Expenses',
 '2c. PMS Expenses',
 '3a. GPMS by Age ',
 '3b. GPMS by Age ',
 '4a. GPMS by Rurality ',
 '4b. GPMS by Rurality ',
 '5a. GPMS by Practice Size',
 '5b. GPMS by Practice Size',
 '6a. GPMS by SHA_GOR',
 '6b. GPMS by SHA_GOR ',
 '7a. GPMS by NHS ER',
 '7b. GPMS by NHS ER',
 '8. GPMS Distribution',
 '9a. All Salaried',
 '9b. Salaried by Country ',
 '9c. Salaried by Age',
 '9d. Salaried by Rurality',
 '9e. Salaried by SHA_GOR',
 '9f. Salaried by NHS ER',
 '10. Salaried Distribution ',
 '11. Combined GPs']
```

In [4]:

```
ws = wb.get_sheet_by_name('1a. GPMS Cash Terms ')
```

In [5]:

```
ws["B7"].value
```

Out[5]:

```
'All Practice Types'
```

In [6]:

```
for i in range(1, 200):
    print(ws["B{}".format(i)].value, ws["C{}".format(i)].value)
```

```
Return to contents None
None None
GP EARNINGS AND EXPENSES 2002/03 TO 2012/13 CASH TERMS
UK, ENGLAND, SCOTLAND, WALES, NORTHERN IRELAND None
None None
GPMS UK None
Practice Type Year
All Practice Types 2002/031
```

None 2003/041
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Dispensing 2002/031
None 2003/041
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Non-Dispensing 2002/031,2
None 2003/041,2
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
None None
None None
GPMS ENGLAND None
Practice Type Year
All Practice Types 2002/03
None 2003/04
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Dispensing 2002/03
None 2003/04
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12

None 2012/13
All Non-Dispensing 2002/032
None 2003/042
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
None None
None None
GPMS SCOTLAND None
Practice Type Year
All Practice Types 2002/034
None 2003/04
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Dispensing 2002/034
None 2003/04
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Non-Dispensing 2002/032,4
None 2003/042
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
None None
None None
GPMS WALES None
Practice Type Year
All Practice Types 2002/03
None 2003/04
None 2004/05
None 2005/06

None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Dispensing 2002/03
None 2003/04
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Non-Dispensing 2002/032
None 2003/042
None 2004/053
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
None None
None None
GPMS NORTHERN IRELAND None
Practice Type Year
All Practice Types 2002/03
None 2003/04
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Dispensing 2002/03
None 2003/04
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
All Non-Dispensing 2002/032
None 2003/042

```
None 2004/05
None 2005/06
None 2006/07
None 2007/08
None 2008/09
None 2009/10
None 2010/11
None 2011/12
None 2012/13
None None
1 GB not UK figures. Despite Northern Ireland figures being pub
lished for these years, UK figures were not published. None
2 Pre 2004/05 two categories were used to describe non dispense
rs,  "non dispensers with help" and "non dispensers without hel
p". None
 Therefore, figures for All Non-Dispensers for these years are
not available. However, this was not the case for Northern Irel
and figures. None
3 In 2004/05, only high level figures were adjusted, therefore
some results are not available. See the definitions and changes
tab  None
 for more information on adjustments. None
4 PMS figures are not available for Scotland in 2002/03, theref
ore the GPMS figures are not included. None
None None
Return to Top None
None None
Copyright © 2014, The Health and Social Care Information Centr
e. All Rights Reserved. None
None None
```

Observe that the date values and the footnote numbers run together to give a single string value. That means some parsing has to be applied to separate it into a `(date, footnote)` pair, whose second member will a list of notes that apply. This is sensible from an analytical point of view, since then the year value can be used for time-based analysis. From an openness point of view it would be much better to have a separate column for the footnotes that should be applied to the row. Then again, from an openness point of view it might be better not to use Excel spreadsheets

Turns out that may not be as useful as I thought. It would probably be easier to maintain the column values as part ofthe processing logic. (*This was borne out when I wrote a non-terminating loop when experimenting with the code below you have Been Warned*)

In [7]:

```
ws["B3"].value
```

Out[7]:

```
'GP EARNINGS AND EXPENSES 2002/03 TO 2012/13 CASH TERMS \nUK, E
NGLAND, SCOTLAND, WALES, NORTHERN IRELAND'
```

When I wrote this I knew it wasn't going to be a YAGNI because of the observations above. Sometimes I'm happy to build bottom-up just to "get to grips" with one small aspect of a problem - get a little hand-to-hand combat with the problem and notch up a small gain before returning to the bigger picture.

In [8]:

```
def year_refs(s):
    """Separate the year string into the year plus the list of reference
s"""
    return s[:7], s[7:].split(",")
```

In [9]:

```
year_refs("2009/101,3,4")
```

Out[9]:

```
('2009/10', ['1', '3', '4'])
```

In [10]:

```
ws["d81"].value
```

Out[10]:

```
'-'
```

In [11]:

```
def num_val(val):
    return 0 if val == "-" else val
```

In [12]:

```
num_val(32.456)
```

Out[12]:

```
32.456
```

In [13]:

```
num_val("-")
```

Out[13]:

```
0
```

In [14]:

```
3 == "banana"
```

Out[14]:

```
False
```

In [15]:

```
cell = ws["B3"]
```

Probably a good idea to look at how we can find the relevant areas in a worksheet, then analyze the content of those areas (which will vary in size, increasing as the years go by, otherwise a static description of the shapes of the sheet might have done.

In [16]:

```
ws["B3"].value # Sheet heading
```

Out[16]:

```
'GP EARNINGS AND EXPENSES 2002/03 TO 2012/13 CASH TERMS \nUK, E
NGLAND, SCOTLAND, WALES, NORTHERN IRELAND'
```

In [17]:

```
ws["B5"].value # Table heading
```

Out[17]:

```
'GPMS UK'
```

In [18]:

```
cells = ws.get_cell_collection()
```

In [19]:

```
from collections import defaultdict

cols_in_row = defaultdict(list)

for cell in cells:
    if cell.value is not None:
        cols_in_row[cell.row].append(cell.column)
```

In [20]:

```
max_row = max(c for c in sorted(cols_in_row.keys()))
max_row
```

Out[20]:

```
198
```

Note that cell J43 has a spurious value that should really be ignored. Wonder how long that's been there and who knows it is? You can correct manually for that sort of stuff, but it's much better to try and think of a rule.

In [21]:

```
cols_in_row[43].remove('J')
cols_in_row[43]
```

Out[21]:

```
['E', 'F', 'C', 'G', 'D', 'B']
```

max_col = max(c for r in range(rows) for c in cols_in_row[r]) max_colcols = ord(max_col)-ord("@")

Initial Visualization

Since Python has some easy graphics, I wanted a quick and dirty way to get an idea, quite literally, of the shape of the problem.

I did this by building a quick and dirty visual mapping of cells as pixels in a graphic.

In [22]:

```
pixels = [] # straight list of pixel values for graphic
matrix = []
#print("  ".join(list("ABCDEFG"))) # Column headings
for row_num in range(max_row):
    cols = cols_in_row[row_num]
    row_string = []
    row_matrix = []
    for col_name in "ABCDEFG":
        row_string.append("*" if col_name in cols else " ")
        row_matrix.append(col_name in cols)
    #print("  ".join(row_string))
    matrix.append(row_matrix)
    pixels += [1-p for p in row_matrix] + [1]*7 # add pixel row plus blank
row
```

In [23]:

```
from PIL import Image
im = Image.new("1", (7, 198*2))
```

In [24]:

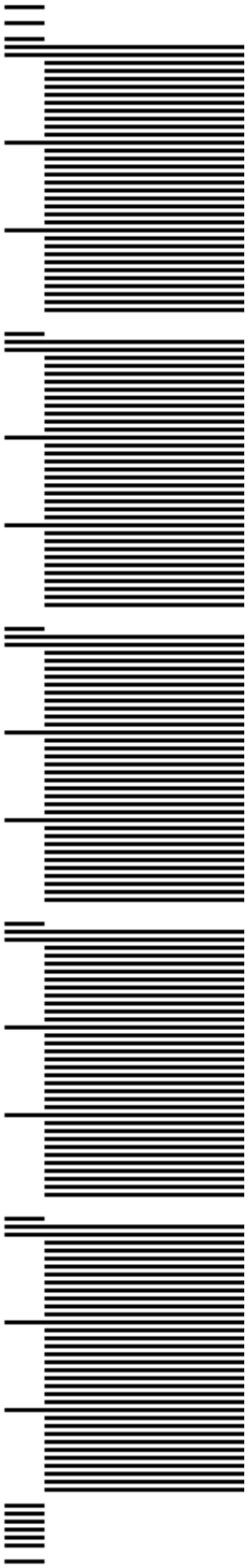
```
im.putdata(pixels)
```

Of course there's a lot of *hard-coded knowledge* in this code, but it provides an initial glimpse as a single worksheet.

In [25]:

```
im.resize((7*20, 198*4))
```

Out[25]:



This visualization makes the pattern of the tables more obvious. Each begins with a row with a single cell, followed by two cells with six rows and a number of rows with five cells. It would be possible to construct a vector with the number of cells in each row, and then search that for patterns characteristic of the start of a table. Whenever you find yourself thinking "pattern," though, it's worth considering using Python's `re` regular expression-based pattern-matching algorithm (another effbot

contribution). Since no row has more than seven cells we can construct a ***string of row lengths*** and then use pattern matching to find the starting positions of the tables. The task then simplifies to finding the string `"1665"`.

In [26]:

```
str_sizes = "".join(str(sum(x for x in row)) for row in matrix) # only work
s for widths up to 9 ...
```

In [27]:

```
import re
for m in re.finditer("(1665)", str_sizes):
    print(m.span()[0])
```

```
5
42
79
116
153
```

Maybe there's some easier way to determing the shape without all these complex manipulations. I don't know about you, but I often find my second approach to a problem is more intelligent than the first (that's why we are often recommended to write a prototype *and then throw it away*). So consider everything above as prototypical, offering insight into the necessary analysis but using a horribly inefficient algorithm.

In [28]:

```
len(ws.columns)
```

Out[28]:

```
16
```

We can form the count of each column by summing the number of value-holding cells.

In [29]:

```
col_counts =[sum(cell.value is not None for cell in column) for column in w
s.columns]
```

Next we eliminate *invalid* columns (those with no more than one value in them - this rule would eliminate the messy little tick mark in column "J").

In [30]:

```
valid_cols = [i for (i, ct) in  enumerate(col_counts) if ct > 1]
valid_cols
```

Out[30]:

```
[1, 2, 3, 4, 5, 6]
```

In [31]:

```
row_counts =[sum(cell.value is not None for cell in row) for row in ws.row
s]
valid_rows = [i for (i, ct) in  enumerate(row_counts) if ct > 0]
len(valid_rows)
```

Out[31]:

```
185
```

In [32]:

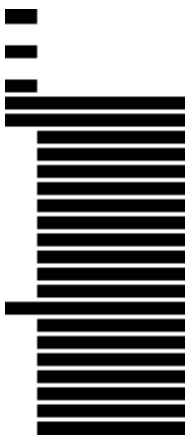
```
ncols = len(valid_cols)
max_row_num = max(valid_rows)
pixels = []
row_counts = []
for row in range(max_row_num):
    cell_strings = []
    row_pixels = []
    for col in valid_cols: # use .extend on a generator expression?
        value = ws.rows[row][col].value
        row_pixels.append(value is None)
    pixels += row_pixels*3 + [1]*ncols # blank line
    row_counts.append(sum(1-pixel for pixel in row_pixels))
```

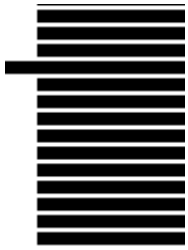
This new representation is "chunkier" - each cell is three rows of pixels, with only a single blank row between cells.

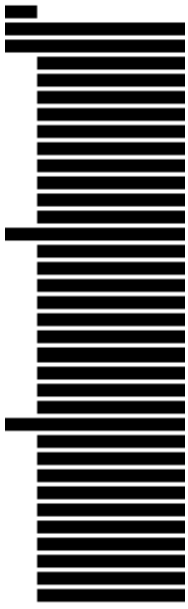
In [33]:

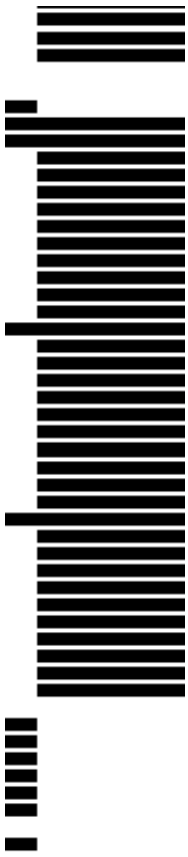
```
im = Image.new("1", (ncols, 4*(max_row_num)))
im.putdata(pixels)
im.resize((14*ncols, 8*(max_row_num+1)))
```

Out[33]:









It appears that this techique can be fairly effective in getting idea of the shape of a worksheet. The next step will be to turn that into a function with the worksheet as a parameter, and apply it to all the sheets on a workbook.

In [34]:

```
%matplotlib inline
```

In [35]:

```
def visualize(ws):
    col_counts =[sum(cell.value is not None for cell in column) for column
in ws.columns]
    valid_cols = [i for (i, ct) in  enumerate(col_counts) if ct > 1]
    row_counts =[sum(cell.value is not None for cell in row) for row in w
s.rows]
    valid_rows = [i for (i, ct) in  enumerate(row_counts) if ct > 0]
    ncols = len(valid_cols)
    max_row_num = max(valid_rows)
    pixels = []
    row_counts = []
    for row in range(max_row_num):
        cell_strings = []
        row_pixels = []
        for col in valid_cols:
            value = ws.rows[row][col].value
            row_pixels.append(value is None)
        pixels += row_pixels*3 + [1]*ncols # blank line
        row_counts.append(sum(1-pixel for pixel in row_pixels))
    im = Image.new("1", (ncols, 4*(max_row_num)))
    im.putdata(pixels)
    return im.resize((14*ncols, 8*(max_row_num+1))), row_counts
```

In [36]:

```
images = []; row_counts = []
for ws in wb.worksheets:
    if ws.sheet_state != "hidden": # exclude hidden sheets
        image, counts = visualize(ws)
        images.append(image)
        row_counts.append(counts)
```

This gives us a list of worksheet images and a list of row-counts. We compute the size of the image necessary to contain all of these individual graphics, and create an image of that size.

In [37]:

```
LEFT_MARGIN = 4
im_width = sum(i.size[0] for i in images)+(len(images)-1)*LEFT_MARGIN
im_height = max(i.size[1] for i in images)
im_width, im_height
```

Out[37]:

```
(3370, 2680)
```

In [38]:

```
len(row_counts)
```

Out[38]:

```
28
```

In [39]:

```
table_widths = [max(c) for c in row_counts]
```

In [40]:

```
x_offset = 0
big_image = Image.new("1", (im_width, im_height), 1)
for i, im in enumerate(images):
    big_image.paste(im, (x_offset, 0))
    x_offset += im.size[0]+LEFT_MARGIN
```

In [41]:

```
big_image
```

Out[41]:

Note that in a live notebook (you may just be reading a PDF copy) the above graphic can be seen at full size and scrolled in two dimensions. For presentation purposes the graphic is automatically scaled to the width of the notebook.

In [42]:

```
ws = wb.worksheets[2]
ws
```

Out[42]:

```
<Worksheet "1a. GPMS Cash Terms ">
```

In [43]:

```
# Page header is in B3 always?
page_header = ws["B3"].value
print(page_header)
```

```
GP EARNINGS AND EXPENSES 2002/03 TO 2012/13 CASH TERMS
UK, ENGLAND, SCOTLAND, WALES, NORTHERN IRELAND
```

In [44]:

```
counts_string = "".join(str(n) for n in row_counts[2])
counts_string
```

Out[44]:

```
'10101665555555555655555555556555555555500166555555555656555555
55565555555555001665555555555655555555556555555555001665555555
55565555555555655555555550016655555555556555555555565555555550
111111010'
```

Looking for the pattern `"1665"*` in `counts_string` finds a six-column table. The first line is the name of the table. The second line is the column headings. The remainder of the table is a number of repeating groups. The first column is special because unchanged values aren't repeated (which is why subsequent lines only have five elements). While this is helpful for the human reader's comprehension it has to be corrected for the computer.

* Yes, this is a fix - that string was chosen because I knew there were six columns in the tables. We may or may not get to the computation of the number of columns later.

In [45]:

```
table_starts = [x.start(0) for x in re.finditer("1665", counts_string)]
table_starts
```

Out[45]:

```
[4, 41, 78, 115, 152]
```

In [46]:

```
table_lens = [x.end("X")-x.start("X") for x in re.finditer("16(?P<X>65[5
6]+)", counts_string)]
table_lens
```

Out[46]:

```
[33, 33, 33, 33, 33]
```

In [47]:

```
number_of_tables = len(table_starts)
number_of_tables
```

Out[47]:

```
5
```

Let's learn how to access the elements we need in order to construct a usable data source. The crucial facts for each table are the number of groups, the number of rows in each group and the number of columns in the table. When you think about it this is simply a description of a three-dimensional structure. The first table reports figures over the whole UK. The remainder analyze that information geographically, adding a fourth dimension to the data.

Let's do a little work on the first table to learn what we'll need to do in the general case.

Firstly, let's see how to access the various "chunks" of the table, beginning with its title.

In [48]:

```
start_row = table_starts[0]
table_len = table_lens[0]
ws["B5"].value, ws["B5"], ws.columns[1][4], ws.columns[1][start_row]
```

Out[48]:

```
('GPMS UK',
 <Cell 1a. GPMS Cash Terms .B5>,
 <Cell 1a. GPMS Cash Terms .B5>,
 <Cell 1a. GPMS Cash Terms .B5>)
```

In [49]:

```
title_cell = ws.columns[1][start_row]
title_cell.value
```

Out[49]:

```
'GPMS UK'
```

Next we need to extract the column names, which are on the row following the title.

In [50]:

```
table_cols = 6
headers = [c.value for c in ws.rows[start_row+1][1:table_cols+1]]
# really should have computed that "6" from the pattern ...
headers
```

Out[50]:

```
['Practice Type',
 'Year',
 'Estimated Population',
 'Gross Earnings',
 'Total Expenses',
 'Income Before Tax']
```

In [51]:

```
# This is a cheesy way to work out how many groups there are
pat = "(?P<X>(65+)+)"
m = re.search(pat, counts_string, start_row+2)
assert m.groups(0)[0].replace(m.groups(0)[1], "") == "" # only true for fix
ed groups
```

In [52]:

```
group_count = len(m.groups(0)[0])//len(m.groups(0)[1])
group_count
```

Out[52]:

```
3
```

In [53]:

```
group_len = table_len//group_count
group_len
```

Out[53]:

```
11
```

So we now seem to have enough information about the tables to be able to built Pandas dataframes from them. Hooray!

In [54]:

```
import pandas as pd
first_data_row = start_row+2
dataframes = []
for data_start_row in range(first_data_row, first_data_row+table_len, grou
p_len):
    group_cells = [[ws.rows[row][col].value for col in range(1, table_col
s+1)]
                   for row in range(data_start_row, data_start_row+group_le
n)]
    group = pd.DataFrame(group_cells, columns=headers)
    group[headers[0]] = group[headers[0]][0]
    dataframes.append(group)
```

In [55]:

```
big_frame = pd.concat(dataframes)
```

In [56]:

```
big_frame.index = range(table_len)
```

In [57]:

```
big_frame
```

Out[57]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	All Practice Types	2002/031	31362	184154	111439	72716
1	All Practice Types	2003/041	31594.98	203613.5	121594.8	82018.71
2	All Practice Types	2004/05	33887.99	230096.7	129926.2	100169.5
3	All Practice Types	2005/06	33874.98	245019.6	135016	110003.6
4	All Practice Types	2006/07	33887	247361.6	139694.4	107667.2
5	All Practice Types	2007/08	33622	251997.3	145925.4	106071.9
6	All Practice Types	2008/09	33371	258600	153300	105300
7	All Practice Types	2009/10	33050	262700	156900	105700
8	All Practice Types	2010/11	33000	266500	162400	104100
9	All Practice Types	2011/12	32950	267900	164900	103000
10	All Practice Types	2012/13	32850	271800	169700	102000
11	All Dispensing	2002/031	4851	271003	183830	87172
12	All Dispensing	2003/041	5233.81	286133.3	188593.1	97540.19
13	All Dispensing	2004/05	5329.23	317973.9	198418.1	119555.8
14	All Dispensing	2005/06	5303.1	331894.9	204833.9	127061
15	All Dispensing	2006/07	5288	330790.8	203794.5	126996.3
16	All Dispensing	2007/08	5121	338498.9	213334.2	125164.7
17	All Dispensing	2008/09	4910	346800	225400	121500
18	All Dispensing	2009/10	4850	348200	226800	121400

	Dispensing					
	All					

Here's a thing. I couldn't work out a quick way to do a linear interpolation backwards in time here. I'm sure somebody smarter can, if it's important.

In [58]:

```
import numpy as np
big_frame["Estimated Population"][22:33].replace("-", np.NaN).interpolate()
```

Out[58]:

```
22          NaN
23          NaN
24     28558.76
25     28571.88
26     28599.00
27     28501.00
28     28461.00
29     28200.00
30     27950.00
31     27950.00
32     27900.00
Name: Estimated Population, dtype: float64
```

In [59]:

```
big_frame
```

Out[59]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	All Practice Types	2002/031	31362	184154	111439	72716
1	All Practice Types	2003/041	31594.98	203613.5	121594.8	82018.71
2	All Practice Types	2004/05	33887.99	230096.7	129926.2	100169.5
3	All Practice Types	2005/06	33874.98	245019.6	135016	110003.6
4	All Practice Types	2006/07	33887	247361.6	139694.4	107667.2
5	All Practice Types	2007/08	33622	251997.3	145925.4	106071.9
6	All Practice Types	2008/09	33371	258600	153300	105300
7	All Practice Types	2009/10	33050	262700	156900	105700

8	All Practice Types	2010/11	33000	266500	162400	104100
9	All Practice Types	2011/12	32950	267900	164900	103000
10	All Practice Types	2012/13	32850	271800	169700	102000
11	All Dispensing	2002/031	4851	271003	183830	87172
12	All Dispensing	2003/041	5233.81	286133.3	188593.1	97540.19
13	All Dispensing	2004/05	5329.23	317973.9	198418.1	119555.8
14	All Dispensing	2005/06	5303.1	331894.9	204833.9	127061
15	All Dispensing	2006/07	5288	330790.8	203794.5	126996.3
16	All Dispensing	2007/08	5121	338498.9	213334.2	125164.7
17	All Dispensing	2008/09	4910	346800	225400	121500
18	All Dispensing	2009/10	4850	348200	226800	121400
	All					

A little excursion into JSON here: note that conversiopn to JSON doesn't necessarily preserve the order of the columns. Fortunately we have already extracted the desired order, so we can use the list of headers to select the columns in the right order.

In [60]:

```
js = big_frame.to_json()
```

In [61]:

```
pd.read_json(js)[headers].sort()
```

Out[61]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	All Practice Types	2002/031	31362	184154	111439	72716
1	All Practice Types	2003/041	31594.98	203613.5	121594.8	82018.71

2	All Practice Types	2004/05	33887.99	230096.7	129926.2	100169.5
3	All Practice Types	2005/06	33874.98	245019.6	135016	110003.6
4	All Practice Types	2006/07	33887	247361.6	139694.4	107667.2
5	All Practice Types	2007/08	33622	251997.3	145925.4	106071.9
6	All Practice Types	2008/09	33371	258600	153300	105300
7	All Practice Types	2009/10	33050	262700	156900	105700
8	All Practice Types	2010/11	33000	266500	162400	104100
9	All Practice Types	2011/12	32950	267900	164900	103000
10	All Practice Types	2012/13	32850	271800	169700	102000
11	All Dispensing	2002/031	4851	271003	183830	87172
12	All Dispensing	2003/041	5233.81	286133.3	188593.1	97540.19
13	All Dispensing	2004/05	5329.23	317973.9	198418.1	119555.8
14	All Dispensing	2005/06	5303.1	331894.9	204833.9	127061
15	All Dispensing	2006/07	5288	330790.8	203794.5	126996.3
16	All Dispensing	2007/08	5121	338498.9	213334.2	125164.7
17	All Dispensing	2008/09	4910	346800	225400	121500
18	All Dispensing	2009/10	4850	348200	226800	121400
	All					

Here I'm experimenting with replacing Nan values. I'm not sure I use the `tf` frame again.

In [62]:

```
tf = pd.read_json(js)[headers].sort()[headers[2:]].replace("-", np.NaN) \
           -big_frame[headers[2:]] .replace("-", np.NaN)>1e-10
```

Given a list of the columns in a dataframe it's easy to select just the columns you want to see.

In [63]:

```
big_frame[headers[2:]]
```

Out[63]:

	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	31362	184154	111439	72716
1	31594.98	203613.5	121594.8	82018.71
2	33887.99	230096.7	129926.2	100169.5
3	33874.98	245019.6	135016	110003.6
4	33887	247361.6	139694.4	107667.2
5	33622	251997.3	145925.4	106071.9
6	33371	258600	153300	105300
7	33050	262700	156900	105700
8	33000	266500	162400	104100
9	32950	267900	164900	103000
10	32850	271800	169700	102000
11	4851	271003	183830	87172
12	5233.81	286133.3	188593.1	97540.19
13	5329.23	317973.9	198418.1	119555.8
14	5303.1	331894.9	204833.9	127061
15	5288	330790.8	203794.5	126996.3
16	5121	338498.9	213334.2	125164.7
17	4910	346800	225400	121500
18	4850	348200	226800	121400
19	5050	356500	237700	118800
20	5000	357800	241900	115900
21	4900	360400	246100	114300
22	-	-	-	-
23	-	-	-	-

24	28558.76	213738.3	117145.2	96593.08
25	28571.88	228319.4	121638.9	106680.5
26	28599	231935.4	127842.2	104093.2
27	28501	236454.8	133813.5	102641.3
28	28461	243400	140900	102500
29	28200	248000	145000	103000
30	27950	250300	148900	101500
31	27950	251900	151200	100700
32	27900	256200	156300	99900

Table Extraction With `pandas`

Data Scientists may rejoin the excursion at this point

Having learned what was necessary to extract the tables, let's see if `pandas` can't make it easier for us.

For a fully general solution using the pattern-matching approach outlined above, we need to be able to cope with wider tables - so far the code just assumed that none of the rows would be more than 9 cells long. This is possible if we define functions to map between row lengths and single-character strings.

In [64]:

```
rl_charset = "0123456789abcdefghijklmnopqrstuvwxyz"

def rl_dig(i):
    return rl_charset[i]

def dig_rl(c):
    return rl_charset.index(c)

# testing, even!
for c in rl_charset:
    assert rl_dig(dig_rl(c)) == c

for i in range(len(rl_charset)):
    assert dig_rl(rl_dig(i)) == i
```

In [65]:

```
def extract_tables(ws):
    import pandas as pd
    if ws.sheet_state == "hidden":
        return []
    col_counts = pd.Series([sum(cell.value is not None for cell in column)
for column in ws.columns])
    valid_cols = [i for (i, ct) in  enumerate(col_counts) if ct > 1]
```

```
    #print("valid columns:", valid_cols)
    row_counts =[sum(row[col_num].value is not None for col_num in valid_co
ls) for row in ws.rows]
    counts_string = "".join(rl_dig(n) for n in row_counts)
    valid_rows = [i for (i, ct) in enumerate(valid_cols) if ct > 0]
    #print("valid rows:", valid_rows)
    table_width = len(valid_cols)
    patchars = rl_dig(table_width), rl_dig(table_width), rl_dig(table_widt
h-1)
    #print(title)
    #print(valid_rows, valid_cols)
    table_starts = [x.start(0) for x in re.finditer("1{}{}{}".format(*patch
ars), counts_string)] # assumes table_length known
    if not table_starts:
        return []
    tables = []
    for start_row in table_starts:
            pat_string = "1{}((?P<X>{}{}+)+)".format(*patchars)
            headers = [c.value for c in ws.rows[start_row+1][1:table_widt
h+1]]
            title = ws.columns[1][start_row].value
            # Normalize indexing by creating a list-of-lists
            # representation of just the "interesting" cells
            sheet_cells = [[ws.rows[row][col] for col in valid_cols] for ro
w in valid_rows]
            #print(pat_string, counts_string[start_row:])
            pat = re.compile(pat_string)
            m = pat.match(counts_string, start_row)
            table_len = len(m.groups(0)[0])
            group_len = len(m.groups(0)[1])
            group_count = table_len//group_len
            #print(title, pat_string, table_len, group_count, group_len)
            first_data_row = start_row+2
            groups = []
            #print("range:", first_data_row, first_data_row+table_len, grou
p_len)
            for group_num in range(group_count):
                data_start_row = first_data_row+group_num*group_len
                #print("starting at:", data_start_row)
                #print("group len:", group_len)
                group_cells = [[c.value for c in ws.rows[row][1:table_widt
h+1]]
                               for row in range(data_start_row, data_star
t_row+group_len)]
                group = pd.DataFrame(group_cells, columns=headers)
                group[headers[0]] = group[headers[0]][0]
                group.index = list(range(group_len))
                groups.append(group.replace("-", np.NaN))
            table = pd.concat(groups)
            table.index = (list(range(len(groups)*group_len)))
            tables.append(table)
    return tables
```

With the above function defined, we can create a list of lists of the dataframes on each sheet.

In [66]:

```
table_sheets = [extract_tables(ws) for ws in wb.worksheets if ws.sheet_stat
e != "hidden"]
```

In [67]:

```
lengths = [len(f) if f else None for f in table_sheets]
titles = [wb.sheetnames[i] for i, s in enumerate(wb.worksheets) if s.shee
t_state != "hidden"]
print(sum(l for t, l in zip(titles, lengths) if l), "tables in all")
```

```
105 tables in all
```

In [68]:

```
no_tables = pd.DataFrame([], columns=["NO TABLES"])
```

Now we forms a list of the first table on each sheet (substituting a little null table of our own if there aren't any). In reading the code later I realised that I could shorten the formatting code by not using an unpacking `for` loop and then starring the tuple to pass it into the `.format()` method. No biggy.

In [69]:

```
flist = [tables[0] if tables else no_tables for tables in table_sheets]
lenlist = [len(tables) for tables in table_sheets]
wlist = [f.shape[1] for f in flist]
from IPython.display import HTML
html = "".join("<h3>{}: {} tables, {} columns</h3><p>{}</p>".format(title,
length, width, table.to_html())
               for (title, length, width, table) in zip(titles, lenlist, wl
ist, flist))
```

Sorry for the length of this output. I decided to leave it fully expanded to make it relatively easy to check against the spreadsheets. It's easy to render HTML in a Jupyter notebook.

In [70]:

```
HTML(html)
```

Out[70]:

Contents: 0 tables, 1 columns

	NO TABLES

Definitions and Changes: 1 tables, 1 columns

	Strategic Health Authorities were crated in 2002 to manage the local NHS on behalf of the Secretary of State. There were originally 28 SHAs,
0	on July 1 2006, this number was reduced to 10.

1	on July 1 2006, this number was reduced to 10.
2	NHS England Regions
3	NHS England Regions

1a. GPMS Cash Terms : 5 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Incon Befor
0	All Practice Types	2002/031	31362.000000	184154.000000	111439.000000	72716
1	All Practice Types	2003/041	31594.976437	203613.465384	121594.750565	82018
2	All Practice Types	2004/05	33887.990000	230096.703402	129926.179861	10016
3	All Practice Types	2005/06	33874.980000	245019.575577	135015.975718	11000
4	All Practice Types	2006/07	33887.000000	247361.596509	139694.391103	10766
5	All Practice Types	2007/08	33622.000000	251997.276539	145925.375805	10607
6	All Practice Types	2008/09	33371.000000	258600.000000	153300.000000	10530
7	All Practice Types	2009/10	33050.000000	262700.000000	156900.000000	10570
8	All Practice Types	2010/11	33000.000000	266500.000000	162400.000000	10410
9	All Practice Types	2011/12	32950.000000	267900.000000	164900.000000	10300
10	All Practice Types	2012/13	32850.000000	271800.000000	169700.000000	10200
	All					

11	Dispensing	2002/031	4851.000000	271003.000000	183830.000000	87172
12	All Dispensing	2003/041	5233.809928	286133.318928	188593.127027	97540
13	All Dispensing	2004/05	5329.230000	317973.891450	198418.141182	11955
14	All Dispensing	2005/06	5303.100000	331894.907881	204833.866238	12706
15	All Dispensing	2006/07	5288.000000	330790.840852	203794.542444	12699
16	All Dispensing	2007/08	5121.000000	338498.927168	213334.193203	12516
17	All Dispensing	2008/09	4910.000000	346800.000000	225400.000000	12150
18	All Dispensing	2009/10	4850.000000	348200.000000	226800.000000	12140
19	All Dispensing	2010/11	5050.000000	356500.000000	237700.000000	11880
20	All Dispensing	2011/12	5000.000000	357800.000000	241900.000000	11590
21	All Dispensing	2012/13	4900.000000	360400.000000	246100.000000	11430
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	28558.760000	213738.267866	117145.183241	96593
25	All Non-Dispensing	2005/06	28571.880000	228319.425268	121638.918111	10668
26	All Non-Dispensing	2006/07	28599.000000	231935.398247	127842.172485	10409
27	All Non-Dispensing	2007/08	28501.000000	236454.841085	133813.500646	10264
28	All Non-Dispensing	2008/09	28461.000000	243400.000000	140900.000000	10250
29	All Non-Dispensing	2009/10	28200.000000	248000.000000	145000.000000	10300
30	All Non-Dispensing	2010/11	27950.000000	250300.000000	148900.000000	10150

31	All Non-Dispensing	2011/12	27950.000000	251900.000000	151200.000000	1007(
32	All Non-Dispensing	2012/13	27900.000000	256200.000000	156300.000000	9990(

1b. GPMS Real Terms: 1 tables, 6 columns

	Country	Year	Estimated Population	Gross Earnings (Real Terms)4	Total Expenses (Real Terms)4	Income Before (Real Te
0	UK	2002/031	31362.000000	232122.823054	140466.866201	91657.2
1	UK	2003/041	31594.976437	251773.206810	150354.988671	101418.
2	UK	2004/05	33887.990000	276869.157482	156336.668110	120531.
3	UK	2005/06	33874.980000	289544.758585	159551.203214	129993.
4	UK	2006/07	33887.000000	284148.750202	160469.479509	123679.
5	UK	2007/08	33622.000000	282369.883738	163513.399693	118856.
6	UK	2008/09	33371.000000	281800.000000	167100.000000	114800.
7	UK	2009/10	33050.000000	278600.000000	166400.000000	112100.
8	UK	2010/11	33000.000000	275500.000000	167900.000000	107600.
9	UK	2011/12	32950.000000	270800.000000	166700.000000	104100.
10	UK	2012/13	32850.000000	271800.000000	169700.000000	102000.
11	England	2002/03	25928.000000	241731.478202	147061.708616	94669.7
12	England	2003/04	26147.424787	262720.939918	157870.262291	104850.
13	England	2004/05	27334.110000	290945.974282	166329.917515	124616.
14	England	2005/06	27436.160000	304368.021367	170108.488058	134259.
15	England	2006/07	27279.000000	299544.307750	171385.907143	128158.
16	England	2007/08	27121.000000	298183.603618	174769.519533	123414.
17	England	2008/09	26712.000000	298700.000000	179300.000000	119400.
18	England	2009/10	26400.000000	295000.000000	178900.000000	116000.
19	England	2010/11	26300.000000	292500.000000	181200.000000	111300.
20	England	2011/12	26350.000000	287400.000000	180100.000000	107200.
21	England	2012/13	26200.000000	289300.000000	184200.000000	105100.
22	Scotland	2002/032,3	NaN	NaN	NaN	NaN
23	Scotland	2003/042	3724.697638	184823.315565	102959.650549	81863.6
24	Scotland	2004/05	3771.000000	207366.902681	107861.365264	99505.5
25	Scotland	2005/06	3668.000000	214273.934675	107188.091537	107085.

26	Scotland	2006/07	3727.000000	213723.358032	110950.201244	102773.
27	Scotland	2007/08	3619.000000	209759.422017	111858.121288	97901.3
28	Scotland	2008/09	3730.000000	205400.000000	111300.000000	94300.0
29	Scotland	2009/10	3750.000000	203900.000000	108900.000000	94900.0
30	Scotland	2010/11	3750.000000	200100.000000	107900.000000	92300.0
31	Scotland	2011/12	3700.000000	193300.000000	103600.000000	89700.0
32	Scotland	2012/13	3700.000000	191300.000000	102600.000000	88800.0
33	Wales	2002/032	1721.000000	212822.320939	130883.420695	81940.1
34	Wales	2003/042	1721.000000	230926.805500	139445.220581	91481.5
35	Wales	2004/05	1737.773145	252909.153963	142577.048338	110204.
36	Wales	2005/06	1717.630000	264953.956582	144189.301040	120764.
37	Wales	2006/07	1773.000000	255853.107349	143541.176386	112311.
38	Wales	2007/08	1759.000000	248192.165485	143573.548807	104618.
39	Wales	2008/09	1785.000000	240800.000000	142000.000000	98800.0
40	Wales	2009/10	1750.000000	241500.000000	142400.000000	99200.0
41	Wales	2010/11	1750.000000	235900.000000	140600.000000	95400.0
42	Wales	2011/12	1750.000000	236200.000000	142000.000000	94300.0
43	Wales	2012/13	1750.000000	233800.000000	142800.000000	91000.0
44	Northern Ireland	2002/032	1067.000000	162510.177395	85872.864919	76637.3
45	Northern Ireland	2003/042	1021.000000	173414.230782	89869.975438	83544.2
46	Northern Ireland	2004/05	1045.000000	208276.050364	98596.276510	109679.
47	Northern Ireland	2005/06	1053.000000	218860.595541	102276.493332	116584.
48	Northern Ireland	2006/07	1108.000000	208943.029729	101749.662960	107193.
49	Northern Ireland	2007/08	1123.000000	202848.483125	100818.107820	102030.
50	Northern Ireland	2008/09	1144.000000	200200.000000	102400.000000	97800.0
51	Northern Ireland	2009/10	1100.000000	200700.000000	103700.000000	96900.0
52	Northern Ireland	2010/11	1150.000000	192000.000000	101000.000000	91000.0

53	Northern Ireland	2011/12	1150.000000	194700.000000	101000.000000	93800.0
54	Northern Ireland	2012/13	1150.000000	191100.000000	99000.000000	92200.0

1c. GMS: 5 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Incon Befor
0	All Practice Types	2002/031	25305.000000	176483.000000	106712.000000	69771
1	All Practice Types	2003/041	22007.000000	190942.215518	113345.342858	77596
2	All Practice Types	2004/05	24384.686518	217097.481067	120775.455169	96322
3	All Practice Types	2005/06	24916.470000	232035.092170	125723.416651	10631
4	All Practice Types	2006/07	23956.000000	233000.460080	129470.097096	10353
5	All Practice Types	2007/08	21340.000000	231524.663410	131200.756997	10032
6	All Practice Types	2008/09	21369.000000	237300.000000	138200.000000	99200
7	All Practice Types	2009/10	21250.000000	241400.000000	141100.000000	10040
8	All Practice Types	2010/11	21300.000000	244600.000000	145600.000000	99000
9	All Practice Types	2011/12	21450.000000	246800.000000	148500.000000	98300
10	All Practice Types	2012/13	21400.000000	249700.000000	152400.000000	97300

11	All Dispensing	2002/031	4067.000000	267225.000000	182074.000000	85152
12	All Dispensing	2003/041	3869.000000	277044.049753	182994.517708	94049
13	All Dispensing	2004/05	4060.936753	309301.648738	192265.379599	11703
14	All Dispensing	2005/06	4103.900000	323149.088755	198258.133003	12489
15	All Dispensing	2006/07	3868.000000	321576.601915	197531.974809	12404
16	All Dispensing	2007/08	3452.000000	323799.759670	202046.969528	12175
17	All Dispensing	2008/09	3358.000000	329900.000000	213400.000000	11650
18	All Dispensing	2009/10	3350.000000	331600.000000	214600.000000	11700
19	All Dispensing	2010/11	3400.000000	339400.000000	224200.000000	11520
20	All Dispensing	2011/12	3400.000000	341900.000000	229400.000000	11250
21	All Dispensing	2012/13	3350.000000	343200.000000	232100.000000	11100
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	20323.749765	198674.638025	106490.904905	92183
25	All Non-Dispensing	2005/06	20812.570000	214068.790814	111420.693355	10264
26	All Non-Dispensing	2006/07	20088.000000	215944.878805	116364.594158	99580
27	All Non-Dispensing	2007/08	17888.000000	213717.550692	117528.958826	96188
28	All Non-Dispensing	2008/09	18011.000000	220100.000000	124200.000000	95900
29	All Non-Dispensing	2009/10	17900.000000	224700.000000	127400.000000	97300
30	All Non-Dispensing	2010/11	17900.000000	226500.000000	130600.000000	95900

31	All Non-Dispensing	2011/12	18050.000000	228900.000000	133200.000000	9570(
32	All Non-Dispensing	2012/13	18050.000000	232300.000000	137500.000000	9480(

1d. PMS: 3 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Incon Befor
0	All Practice Types	2002/031	6057.000000	216202.000000	131182.000000	8501(
1	All Practice Types	2003/041	9587.976437	232697.412101	140529.373174	9216(
2	All Practice Types	2004/05	9503.298730	263570.228862	153406.190473	1101(
3	All Practice Types	2005/06	8958.500000	281133.873894	160861.722147	1202(
4	All Practice Types	2006/07	9931.000000	285208.641675	166709.456086	1184(
5	All Practice Types	2007/08	12282.000000	287568.483523	171509.430956	1160(
6	All Practice Types	2008/09	12002.000000	296500.000000	180200.000000	1163(
7	All Practice Types	2009/10	11800.000000	300900.000000	185500.000000	1153(
8	All Practice Types	2010/11	11650.000000	306600.000000	193200.000000	1134(
9	All Practice Types	2011/12	11500.000000	307300.000000	195700.000000	1116(
10	All Practice Types	2012/13	11450.000000	313100.000000	202300.000000	1108(

11	All Dispensing	2002/031	784.000000	290599.000000	192945.000000	97654
12	All Dispensing	2003/041	1364.809928	311899.824372	204464.215746	10743
13	All Dispensing	2004/05	1268.288636	345731.979001	218119.050457	12761
14	All Dispensing	2005/06	1199.200000	361148.166935	226879.733237	13426
15	All Dispensing	2006/07	1420.000000	358921.038261	223374.693450	13554
16	All Dispensing	2007/08	1669.000000	368901.279596	236679.607298	13222
17	All Dispensing	2008/09	1552.000000	383400.000000	251200.000000	13220
18	All Dispensing	2009/10	1500.000000	384900.000000	253800.000000	13110
19	All Dispensing	2010/11	1600.000000	392400.000000	266000.000000	12640
20	All Dispensing	2011/12	1600.000000	391700.000000	268500.000000	12320
21	All Dispensing	2012/13	1550.000000	398000.000000	276600.000000	12130
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	8235.010094	250914.824883	143439.609024	10747
25	All Non-Dispensing	2005/06	7759.300000	268767.325154	150658.461124	11810
26	All Non-Dispensing	2006/07	8511.000000	272910.250986	157255.263035	11565
27	All Non-Dispensing	2007/08	10613.000000	274778.090924	161260.771358	11351
28	All Non-Dispensing	2008/09	10450.000000	283600.000000	169700.000000	11390
29	All Non-Dispensing	2009/10	10300.000000	288500.000000	175500.000000	11300
30	All Non-Dispensing	2010/11	10050.000000	292800.000000	181400.000000	11140

31	All Non-Dispensing	2011/12	9900.000000	293700.000000	184000.000000	10970
32	All Non-Dispensing	2012/13	9900.000000	299800.000000	190700.000000	10920

2a. GPMS Expenses: 5 tables, 11 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Emplo
0	All Practice Types	2002/036	113063.000000	12410.000000	10207.000000	56602.
1	All Practice Types	2003/046	124126.249558	13256.745043	11288.493023	63564.
2	All Practice Types	2004/05	135292.356942	12999.321029	12066.975070	73504.
3	All Practice Types	2005/06	141638.544745	12919.043212	13229.459241	78877.
4	All Practice Types	2006/07	139694.391103	12745.370660	14398.504319	81700.
5	All Practice Types	2007/08	145925.375804	13071.812009	15622.303963	86183.
6	All Practice Types	2008/09	153300.000000	13600.000000	17400.000000	90400.
7	All Practice Types	2009/10	156900.000000	14000.000000	18500.000000	94100.
8	All Practice Types	2010/11	162400.000000	14500.000000	19600.000000	97100.
9	All Practice Types	2011/12	164900.000000	15000.000000	20600.000000	98600.
10	All Practice Types	2012/13	169700.000000	15500.000000	22100.000000	10160

11	All Dispensing	2002/036	185405.000000	13124.000000	9896.000000	60233.(
12	All Dispensing	2003/046	191313.084111	14040.720267	10801.934728	66278.1
13	All Dispensing	2004/05	205902.638392	14406.314503	11594.905238	77174.(
14	All Dispensing	2005/06	213404.084320	14059.245775	12583.766767	83133.2
15	All Dispensing	2006/07	203794.542443	13701.586199	13386.354202	87351.1
16	All Dispensing	2007/08	213334.193202	14307.836257	15054.207893	94487.5
17	All Dispensing	2008/09	225400.000000	15100.000000	16600.000000	100200
18	All Dispensing	2009/10	226800.000000	15700.000000	18200.000000	103400
19	All Dispensing	2010/11	237700.000000	17400.000000	18900.000000	109000
20	All Dispensing	2011/12	241900.000000	18600.000000	20000.000000	112400
21	All Dispensing	2012/13	246100.000000	18800.000000	21400.000000	117900
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	122116.069519	12736.767899	12155.066034	72819.9
25	All Non-Dispensing	2005/06	128318.461049	12707.415268	13349.303359	78087.(
26	All Non-Dispensing	2006/07	127842.172484	12568.564905	14585.652465	80655.1
27	All Non-Dispensing	2007/08	133813.500645	12849.725760	15724.378275	84691.2
28	All Non-Dispensing	2008/09	140900.000000	13300.000000	17500.000000	88700.(
29	All Non-Dispensing	2009/10	145000.000000	13700.000000	18500.000000	92500.(
30	All Non-Dispensing	2010/11	148900.000000	14000.000000	19700.000000	95000.(

31	All Non-Dispensing	2011/12	151200.000000	14300.000000	20700.000000	96100.(
32	All Non-Dispensing	2012/13	156300.000000	14900.000000	22200.000000	98700.(

2b. GMS Expenses: 5 tables, 11 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Employ
0	All Practice Types	2002/036	108237.000000	12096.000000	9809.000000	53201.(
1	All Practice Types	2003/046	115600.525131	12683.857430	10400.755978	57738.2
2	All Practice Types	2004/05	125863.524042	12321.602549	11281.286129	66688.(
3	All Practice Types	2005/06	132185.234434	12298.513575	12493.496228	72049.3
4	All Practice Types	2006/07	129470.097096	12101.807765	13523.490254	74405.(
5	All Practice Types	2007/08	131200.756996	12171.666427	14444.514155	75230.1
6	All Practice Types	2008/09	138200.000000	12600.000000	15900.000000	79500.(
7	All Practice Types	2009/10	141100.000000	12900.000000	16700.000000	82600.(
8	All Practice Types	2010/11	145600.000000	13500.000000	17800.000000	84700.(
9	All Practice Types	2011/12	148500.000000	14000.000000	18700.000000	86400.(
10	All Practice	2012/13	152400.000000	14400.000000	20100.000000	89100.(

	Types					
11	All Dispensing	2002/036	183555.000000	13095.000000	9853.000000	58131.(
12	All Dispensing	2003/046	185604.176630	13689.070881	9780.511597	63242.(
13	All Dispensing	2004/05	199391.930517	14191.693014	10899.540747	73335.(
14	All Dispensing	2005/06	206740.671610	13826.478542	11869.358946	78924.(
15	All Dispensing	2006/07	197531.974809	13194.727787	12674.119703	82800.9
16	All Dispensing	2007/08	202046.969531	13586.262416	13721.796973	85718.3
17	All Dispensing	2008/09	213400.000000	14600.000000	15000.000000	92000.(
18	All Dispensing	2009/10	214600.000000	15100.000000	16200.000000	95100.(
19	All Dispensing	2010/11	224200.000000	16800.000000	17000.000000	100300
20	All Dispensing	2011/12	229400.000000	18000.000000	18300.000000	103500
21	All Dispensing	2012/13	232100.000000	18200.000000	20100.000000	108200
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	111171.638356	11947.935328	11357.563581	65359.1
25	All Non-Dispensing	2005/06	117484.115897	11997.223757	12616.565940	70693.1
26	All Non-Dispensing	2006/07	116364.594159	11891.362990	13687.038905	72789.1
27	All Non-Dispensing	2007/08	117528.958826	11898.679769	14583.983056	73206.2
28	All Non-Dispensing	2008/09	124200.000000	12300.000000	16000.000000	77200.(
29	All Non-Dispensing	2009/10	127400.000000	12500.000000	16800.000000	80300.(

30	All Non-Dispensing	2010/11	130600.000000	12900.000000	17900.000000	81800.(
31	All Non-Dispensing	2011/12	133200.000000	13200.000000	18800.000000	83200.(
32	All Non-Dispensing	2012/13	137500.000000	13700.000000	20100.000000	85500.(

2c. PMS Expenses: 3 tables, 11 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Employ
0	All Practice Types	2002/036	133221.000000	13724.000000	11867.000000	70808.(
1	All Practice Types	2003/046	143695.093797	14571.677112	13326.089722	76937.1
2	All Practice Types	2004/05	159486.037889	14738.297663	14082.994604	90995.1
3	All Practice Types	2005/06	167931.399476	14644.950143	15276.423868	97868.9
4	All Practice Types	2006/07	166709.456088	14446.818280	16690.543538	100810
5	All Practice Types	2007/08	171509.430955	14635.816789	17668.716152	105214
6	All Practice Types	2008/09	180200.000000	15300.000000	20100.000000	109900
7	All Practice Types	2009/10	185500.000000	15900.000000	21600.000000	114800
8	All Practice Types	2010/11	193200.000000	16300.000000	23000.000000	119700
9	All Practice Types	2011/12	195700.000000	16800.000000	24100.000000	121200

10	All Practice Types	2012/13	202300.000000	17500.000000	25800.000000	124900
11	All Dispensing	2002/036	195006.000000	13274.000000	10120.000000	71139.(
12	All Dispensing	2003/046	207496.849040	15037.585432	13697.492572	74883.2
13	All Dispensing	2004/05	226750.039777	15093.563984	13821.436793	89464.4
14	All Dispensing	2005/06	236207.602852	14855.821364	15028.611883	97538.2
15	All Dispensing	2006/07	223374.693458	15256.739648	15549.729458	100895
16	All Dispensing	2007/08	236679.607298	15800.270581	17810.039233	112624
17	All Dispensing	2008/09	251200.000000	16300.000000	20300.000000	117800
18	All Dispensing	2009/10	253800.000000	17100.000000	22700.000000	121800
19	All Dispensing	2010/11	266000.000000	18700.000000	22900.000000	127400
20	All Dispensing	2011/12	268500.000000	19700.000000	23500.000000	131300
21	All Dispensing	2012/13	276600.000000	19900.000000	24300.000000	139100
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	149126.588630	14683.582461	14123.277590	91231.5
25	All Non-Dispensing	2005/06	157379.310617	14612.359991	15314.723216	97920.(
26	All Non-Dispensing	2006/07	157255.263036	14311.688643	16880.880278	100796
27	All Non-Dispensing	2007/08	161260.771357	14452.694827	17646.491689	104049
28	All Non-Dispensing	2008/09	169700.000000	15200.000000	20000.000000	108700
	All Non-					

29	Dispensing	2009/10	175500.000000	15700.000000	21500.000000	113800
30	All Non-Dispensing	2010/11	181400.000000	16000.000000	23000.000000	118500
31	All Non-Dispensing	2011/12	184000.000000	16300.000000	24200.000000	119600
32	All Non-Dispensing	2012/13	190700.000000	17100.000000	26000.000000	122700

3a. GPMS by Age : 4 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before
0	All Practice Types	2002/031	8236.000000	161489.000000	97656.000000	63832.(
1	All Practice Types	2003/041	6470.009398	173784.335542	103632.699889	70151.(
2	All Practice Types	2004/05	6525.640000	199693.408457	111938.971317	87754.4
3	All Practice Types	2005/06	6101.480000	211451.623824	115558.853990	95892.1
4	All Practice Types	2006/07	7042.000000	216009.873374	121079.831389	94930.(
5	All Practice Types	2007/08	6762.000000	219469.363106	126386.285203	93083.(
6	All Practice Types	2008/09	6621.250000	224300.000000	132000.000000	92200.(
7	All Practice Types	2009/10	6550.000000	227300.000000	135100.000000	92300.(
8	All Practice Types	2010/11	6500.000000	230100.000000	139200.000000	90900.(
9	All Practice Types	2011/12	6650.000000	231500.000000	140600.000000	90900.(

10	All Practice Types	2012/13	6700.000000	237900.000000	146800.000000	91100.(
11	All Dispensing	2002/031	1316.000000	239391.000000	161106.000000	78285.(
12	All Dispensing	2003/041	1104.766846	247698.372832	162243.053103	85455.3
13	All Dispensing	2004/053	1038.660000	NaN	176103.900531	NaN
14	All Dispensing	2005/06	942.870000	292088.525086	178381.818476	113706
15	All Dispensing	2006/07	1068.000000	295252.499873	180119.976772	115132
16	All Dispensing	2007/08	958.000000	301620.175545	189300.786347	112319
17	All Dispensing	2008/09	910.580000	298500.000000	192100.000000	106500
18	All Dispensing	2009/10	850.000000	304800.000000	198500.000000	106200
19	All Dispensing	2010/11	850.000000	312800.000000	207600.000000	105100
20	All Dispensing	2011/12	850.000000	317600.000000	213600.000000	104000
21	All Dispensing	2012/13	900.000000	321900.000000	219500.000000	102400
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/053	5486.980000	NaN	99792.846240	NaN
25	All Non-Dispensing	2005/06	5158.610000	196239.643757	103727.623460	92512.(
26	All Non-Dispensing	2006/07	5974.000000	201843.297359	110524.947681	91318.3
27	All Non-Dispensing	2007/08	5804.000000	205909.666637	116001.706964	89907.9
28	All Non-Dispensing	2008/09	5710.670000	224300.000000	132000.000000	92200.(
	All Non-					

29	Dispensing	2009/10	5700.000000	215500.000000	125400.000000	90100.(
30	All Non-Dispensing	2010/11	5650.000000	217600.000000	128900.000000	88700.(
31	All Non-Dispensing	2011/12	5750.000000	218500.000000	129600.000000	88900.(
32	All Non-Dispensing	2012/13	5850.000000	225200.000000	135800.000000	89400.(

3b. GPMS by Age : 4 tables, 11 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Emplo
0	All Practice Types	2002/036	98705.000000	11085.000000	9299.000000	49737.(
1	All Practice Types	2003/046	105353.796471	11572.454910	9907.163876	54213.4
2	All Practice Types	2004/05	116446.793004	11378.284969	10607.655379	63448.(
3	All Practice Types	2005/06	121150.189184	11319.444976	11970.491443	67671.9
4	All Practice Types	2006/07	113993.631264	10800.767493	12693.943448	65961.1
5	All Practice Types	2007/08	126386.285201	11707.716053	14346.866266	74952.(
6	All Practice Types	2008/09	132000.000000	12000.000000	16100.000000	77800.(
7	All Practice Types	2009/10	135100.000000	12300.000000	17400.000000	80800.(
8	All Practice Types	2010/11	139200.000000	12600.000000	18400.000000	83300.(
	All					

9	Practice Types	2011/12	140600.000000	12900.000000	19000.000000	83800.(
10	All Practice Types	2012/13	146800.000000	13400.000000	20500.000000	87700.(
11	All Dispensing	2002/036	162215.000000	12117.000000	8870.000000	54177.(
12	All Dispensing	2003/046	164242.907739	12673.781899	9407.753993	58410.(
13	All Dispensing	2004/05	182400.946700	13705.651086	10357.083551	69556.8
14	All Dispensing	2005/06	185858.004889	12676.902076	11796.392546	73617.1
15	All Dispensing	2006/07	177083.914949	12379.010397	12367.078462	75021.(
16	All Dispensing	2007/08	189300.786326	13116.214186	14103.395010	84720.2
17	All Dispensing	2008/09	192100.000000	13200.000000	15400.000000	86800.(
18	All Dispensing	2009/10	198500.000000	14200.000000	17600.000000	90800.(
19	All Dispensing	2010/11	207600.000000	15200.000000	18100.000000	96000.(
20	All Dispensing	2011/12	213600.000000	16400.000000	19100.000000	100000
21	All Dispensing	2012/13	219500.000000	17200.000000	20300.000000	106600
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	103962.096188	10937.729542	10655.087013	62292.4
25	All Non-Dispensing	2005/06	109323.154732	11071.334423	12002.312540	66585.1
26	All Non-Dispensing	2006/07	102357.211005	10509.675162	12754.230676	64289.9
27	All Non-Dispensing	2007/08	116001.706966	11475.231351	14387.053286	73339.(
	All Non-					

28	Dispensing	2008/09	122500.000000	11800.000000	16200.000000	76400.(
29	All Non-Dispensing	2009/10	125400.000000	12000.000000	17300.000000	79300.(
30	All Non-Dispensing	2010/11	128900.000000	12200.000000	18400.000000	81400.(
31	All Non-Dispensing	2011/12	129600.000000	12400.000000	19000.000000	81400.(
32	All Non-Dispensing	2012/13	135800.000000	12900.000000	20500.000000	84800.(

4a. GPMS by Rurality : 2 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before
0	All Practice Types	2002/031	NaN	NaN	NaN	NaN
1	All Practice Types	2003/041	7311.195003	234464.539428	148259.274389	86205.2
2	All Practice Types	2004/05	6070.930000	278223.722996	169633.250897	108590
3	All Practice Types	2005/06	6100.890000	293481.449644	176514.464042	116966
4	All Practice Types	2006/07	6152.000000	291458.560569	176149.786801	115308
5	All Practice Types	2007/08	6036.000000	297432.772686	183862.586948	113570
6	All Practice Types	2008/09	6174.040000	300500.000000	190700.000000	109800
7	All Practice Types	2009/10	6100.000000	302600.000000	192000.000000	110600
8	All Practice Types	2010/11	6150.000000	306700.000000	198400.000000	108200

9	All Practice Types	2011/12	5950.000000	309300.000000	202600.000000	10670(
10	All Practice Types	2012/13	7400.000000	283300.000000	182000.000000	10130(
11	All Dispensing	2002/031	NaN	NaN	NaN	NaN
12	All Dispensing	2003/041	3739.453937	297079.699915	197962.275944	99117.4
13	All Dispensing	2004/053	3473.000000	NaN	214755.893598	NaN
14	All Dispensing	2005/06	3507.570000	351600.469124	222473.887783	12912(
15	All Dispensing	2006/07	3528.000000	348033.739881	218643.404128	12939(
16	All Dispensing	2007/08	3369.000000	359895.443212	231168.174667	12872(
17	All Dispensing	2008/09	3336.740000	365200.000000	241900.000000	12330(
18	All Dispensing	2009/10	3300.000000	364800.000000	241800.000000	12300(
19	All Dispensing	2010/11	3350.000000	372500.000000	252800.000000	11960(
20	All Dispensing	2011/12	3300.000000	373600.000000	257200.000000	11640(
21	All Dispensing	2012/13	3200.000000	373900.000000	259000.000000	11490(
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/053	2597.930000	NaN	109311.788000	NaN
25	All Non-Dispensing	2005/06	2593.320000	214290.993971	113904.635525	10038(
26	All Non-Dispensing	2006/07	2624.000000	215392.542043	119016.600093	96375.9
27	All Non-Dispensing	2007/08	2667.000000	218528.859302	124105.359715	94423.4

28	All Non-Dispensing	2008/09	2837.300000	224400.000000	130400.000000	94000.(
29	All Non-Dispensing	2009/10	2800.000000	229400.000000	133300.000000	96100.(
30	All Non-Dispensing	2010/11	2800.000000	227000.000000	132600.000000	94500.(
31	All Non-Dispensing	2011/12	2700.000000	230900.000000	136000.000000	94900.(
32	All Non-Dispensing	2012/13	4200.000000	214200.000000	123200.000000	91000.(

4b. GPMS by Rurality : 2 tables, 11 columns

	Practice Type	**Year**	**Total Expenses1**	**Office & General Business2**	**Premises**	**Employ**
0	All Practice Types	2002/036	NaN	NaN	NaN	NaN
1	All Practice Types	2003/046	150721.716439	13081.391933	9900.988597	60601.9
2	All Practice Types	2004/05	176145.912598	13902.309535	11003.743285	72769.4
3	All Practice Types	2005/06	184655.956739	13563.338980	12169.604233	78911.8
4	All Practice Types	2006/07	176149.786801	13252.554070	13171.034813	82129.(
5	All Practice Types	2007/08	183862.586950	13569.979569	14754.365345	87947.8
6	All Practice Types	2008/09	190700.000000	14200.000000	16200.000000	91400.(
7	All Practice Types	2009/10	192000.000000	14500.000000	17100.000000	94300.(
8	All Practice	2010/11	198400.000000	15400.000000	18000.000000	97600.(

	Types					
9	All Practice Types	2011/12	202600.000000	16400.000000	18700.000000	100900
10	All Practice Types	2012/13	182000.000000	14600.000000	18800.000000	94500.
11	All Dispensing	2002/036	NaN	NaN	NaN	NaN
12	All Dispensing	2003/046	200770.247410	14284.454773	10291.085701	65728.
13	All Dispensing	2004/05	222405.383936	14808.225269	11389.350386	77522.
14	All Dispensing	2005/06	231433.488070	14445.037918	12158.311988	84388.
15	All Dispensing	2006/07	218643.404127	14047.093294	12989.647120	88012.
16	All Dispensing	2007/08	231168.174669	14504.399697	14758.266141	96542.
17	All Dispensing	2008/09	241900.000000	15500.000000	16200.000000	101600
18	All Dispensing	2009/10	241800.000000	16000.000000	17100.000000	105000
19	All Dispensing	2010/11	252800.000000	17500.000000	18200.000000	110300
20	All Dispensing	2011/12	257200.000000	18800.000000	18500.000000	113900
21	All Dispensing	2012/13	259000.000000	18500.000000	20400.000000	117900
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	114304.937073	12691.255645	10488.252758	66415.
25	All Non-Dispensing	2005/06	121387.457067	12370.805570	12184.877451	71504.
26	All Non-Dispensing	2006/07	119016.600091	12184.286395	13414.912778	74218.
	All Non-					

27	Dispensing	2007/08	124105.359719	12389.604087	14749.437792	77090.3
28	All Non-Dispensing	2008/09	130400.000000	12700.000000	16300.000000	79300.(
29	All Non-Dispensing	2009/10	133300.000000	12700.000000	17000.000000	81900.(
30	All Non-Dispensing	2010/11	132600.000000	12800.000000	17800.000000	82300.(
31	All Non-Dispensing	2011/12	136000.000000	13400.000000	19000.000000	85000.(
32	All Non-Dispensing	2012/13	123200.000000	11600.000000	17600.000000	76700.(

5a. GPMS by Practice Size: 4 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before T
0	All Practice Types	2002/031	2716.00	226398.000000	143690.000000	82707.00
1	All Practice Types	2003/041	NaN	NaN	NaN	NaN
2	All Practice Types	2004/05	2473.81	298257.747599	178853.606209	119404.1
3	All Practice Types	2005/06	2357.22	315609.229215	183599.604271	132009.6
4	All Practice Types	2006/07	2111.00	309758.259825	184165.698702	125592.5
5	All Practice Types	2007/08	1889.00	310308.150985	186123.620551	124184.5
6	All Practice Types	2008/09	1417.34	303800.000000	183000.000000	120800.(
7	All Practice Types	2009/10	1250.00	308300.000000	187500.000000	120800.(
	All					

8	Practice Types	2010/11	1150.00	314200.000000	193900.000000	120400.0
9	All Practice Types	2011/12	900.00	310300.000000	194300.000000	116000.0
10	All Practice Types	2012/13	800.00	308300.000000	196200.000000	112000.0
11	All Dispensing	2002/031	247.00	383446.000000	275927.000000	107520.0
12	All Dispensing	2003/041	NaN	NaN	NaN	NaN
13	All Dispensing	2004/053	207.77	NaN	314923.123633	NaN
14	All Dispensing	2005/06	182.71	488586.106954	328393.542682	160192.5
15	All Dispensing	2006/07	165.00	457668.013503	296225.891877	161442.1
16	All Dispensing	2007/08	135.00	453254.823064	304414.803351	148840.0
17	All Dispensing	2008/09	94.96	468100.000000	312400.000000	155700.0
18	All Dispensing	2009/10	100.00	453900.000000	305400.000000	148400.0
19	All Dispensing	2010/11	100.00	448700.000000	310500.000000	138200.0
20	All Dispensing	2011/12	50.00	442000.000000	319600.000000	122400.0
21	All Dispensing	2012/13	50.00	435800.000000	317100.000000	118800.0
22	All Non-Dispensing	2002/031,2	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/041,2	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/053	2266.04	NaN	166377.584763	NaN
25	All Non-Dispensing	2005/06	2174.51	294147.934672	167629.967888	126517.9
26	All Non-Dispensing	2006/07	1946.00	288762.104499	169489.005012	119273.0

27	All Non-Dispensing	2007/08	1754.00	299305.984084	177019.111038	122286.8
28	All Non-Dispensing	2008/09	1322.38	292000.000000	173700.000000	118200.0
29	All Non-Dispensing	2009/10	1200.00	296900.000000	178300.000000	118600.0
30	All Non-Dispensing	2010/11	1100.00	304200.000000	185100.000000	119100.0
31	All Non-Dispensing	2011/12	850.00	301800.000000	186100.000000	115600.0
32	All Non-Dispensing	2012/13	750.00	298200.000000	186700.000000	111500.0

5b. GPMS by Practice Size: 4 tables, 11 columns

	Practice Type	**Year**	**Total Expenses1**	**Office & General Business2**	**Premises**	**Employ**
0	All Practice Types	2002/036	149274.000000	16372.000000	12967.000000	73637.0
1	All Practice Types	2003/046	NaN	NaN	NaN	NaN
2	All Practice Types	2004/05	187130.943937	18530.578407	16847.627004	102286
3	All Practice Types	2005/06	192769.394558	18078.945368	18006.222202	107580
4	All Practice Types	2006/07	184165.698697	17290.269796	19432.500071	106730
5	All Practice Types	2007/08	186123.620551	16000.918719	19038.320725	107862
6	All Practice Types	2008/09	183000.000000	17100.000000	20500.000000	105900
7	All Practice Types	2009/10	187500.000000	17100.000000	20500.000000	110800

8	All Practice Types	2010/11	193900.000000	17800.000000	23200.000000	11440(
9	All Practice Types	2011/12	194300.000000	19000.000000	24400.000000	11250(
10	All Practice Types	2012/13	196200.000000	18900.000000	26100.000000	11260(
11	All Dispensing	2002/036	282077.000000	17180.000000	11698.000000	79215.(
12	All Dispensing	2003/046	NaN	NaN	NaN	NaN
13	All Dispensing	2004/05	327166.082303	20517.983203	16709.397411	11236(
14	All Dispensing	2005/06	339036.957693	19412.980570	17185.784577	12909(
15	All Dispensing	2006/07	296225.891879	15177.901394	19132.579636	12209(
16	All Dispensing	2007/08	304414.803333	13715.073259	20384.780741	13135(
17	All Dispensing	2008/09	312400.000000	15900.000000	24300.000000	13540(
18	All Dispensing	2009/10	305400.000000	16200.000000	21100.000000	13750(
19	All Dispensing	2010/11	310500.000000	15400.000000	25000.000000	13620(
20	All Dispensing	2011/12	319600.000000	17700.000000	25400.000000	14080(
21	All Dispensing	2012/13	317100.000000	18500.000000	24600.000000	14300(
22	All Non-Dispensing	2002/036,7	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2003/046,7	NaN	NaN	NaN	NaN
24	All Non-Dispensing	2004/05	174291.320321	18348.356075	16860.301080	10136(
25	All Non-Dispensing	2005/06	180479.477997	17966.855025	18075.158266	10577(
26	All Non-	2006/07	174664.192081	17469.376059	19457.930118	10542(

	Dispensing					
27	All Non-Dispensing	2007/08	177019.111043	16176.853233	18934.687828	106054
28	All Non-Dispensing	2008/09	173700.000000	17200.000000	20200.000000	103800
29	All Non-Dispensing	2009/10	178300.000000	17200.000000	20400.000000	108700
30	All Non-Dispensing	2010/11	185100.000000	18000.000000	23100.000000	112800
31	All Non-Dispensing	2011/12	186100.000000	19100.000000	24300.000000	110700
32	All Non-Dispensing	2012/13	186700.000000	18900.000000	26200.000000	110200

6a. GPMS by SHA_GOR: 10 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Ta
0	All Practice Types	2004/05	1435.42	225213.059692	124993.170700	100219.88
1	All Practice Types	2005/06	1433.63	243222.305295	131227.980462	111994.32
2	All Practice Types	2006/07	1428.00	238946.882563	132445.093445	106501.78
3	All Practice Types	2007/08	1389.00	254165.662808	145908.755191	108256.90
4	All Practice Types	2008/092	1126.63	253100.000000	146700.000000	106400.00
5	All Practice Types	2009/10	1050.00	257200.000000	151500.000000	105700.00
6	All Practice Types	2010/11	1050.00	265800.000000	160300.000000	105500.00
7	All Practice Types	2011/12	1050.00	266000.000000	162300.000000	103800.00

8	All Practice Types	2012/13	1050.00	269200.000000	166500.000000	102700.00
9	All Dispensing	2004/051	171.80	NaN	141279.956003	NaN
10	All Dispensing	2005/06	164.66	273169.213213	152959.942832	120209.27
11	All Dispensing	2006/07	169.00	253452.147049	136739.747700	116712.39
12	All Dispensing	2007/08	168.00	NaN	NaN	NaN
13	All Dispensing	2008/092	100.60	NaN	NaN	NaN
14	All Dispensing	2009/10	100.00	290400.000000	176700.000000	113800.00
15	All Dispensing	2010/11	100.00	283100.000000	175100.000000	108100.00
16	All Dispensing	2011/12	100.00	269700.000000	169900.000000	99800.000
17	All Dispensing	2012/13	100.00	273800.000000	172500.000000	101200.00
18	All Non-Dispensing	2004/051	1263.62	NaN	122778.842250	NaN
19	All Non-Dispensing	2005/06	1268.97	234960.297676	125060.149637	109900.14
20	All Non-Dispensing	2006/07	1259.00	236999.789870	131868.606896	105131.18
21	All Non-Dispensing	2007/08	1221.00	NaN	NaN	NaN
22	All Non-Dispensing	2008/092	1026.03	NaN	NaN	NaN
23	All Non-Dispensing	2009/10	1000.00	254300.000000	149200.000000	105000.00
24	All Non-Dispensing	2010/11	950.00	263700.000000	158500.000000	105200.00
25	All Non-Dispensing	2011/12	950.00	265600.000000	161300.000000	104300.00
26	All Non-Dispensing	2012/13	950.00	268700.000000	165800.000000	102800.00

6b. GPMS by SHA_GOR : 10 tables, 11 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Employee
0	All Practice Types	2004/05	129272.226416	10069.666646	11667.829011	79189.252
1	All Practice Types	2005/06	140704.654876	9610.843921	13326.584774	88776.885
2	All Practice Types	2006/07	132445.093445	8937.379461	14797.153683	88656.044
3	All Practice Types	2007/08	145908.755176	9481.356350	17162.808567	95683.336
4	All Practice Types	2008/09	146700.000000	9500.000000	18400.000000	96000.000
5	All Practice Types	2009/10	151500.000000	10000.000000	19300.000000	100100.00
6	All Practice Types	2010/11	160300.000000	10100.000000	21900.000000	104200.00
7	All Practice Types	2011/12	162300.000000	10200.000000	23300.000000	102700.00
8	All Practice Types	2012/13	166500.000000	10800.000000	25900.000000	110100.00
9	All Dispensing	2004/05	145918.321886	9904.745227	10804.254540	69702.410
10	All Dispensing	2005/06	168095.611563	8616.281793	9936.875380	83841.425
11	All Dispensing	2006/07	136739.747692	7641.191243	12204.018107	77257.754
12	All Dispensing	2007/08	NaN	NaN	NaN	NaN
13	All Dispensing	2008/09	NaN	NaN	NaN	NaN

14	All Dispensing	2009/10	176700.000000	8600.000000	15100.000000	99200.000
15	All Dispensing	2010/11	175100.000000	9100.000000	15900.000000	99800.000
16	All Dispensing	2011/12	169900.000000	10200.000000	19100.000000	94300.000
17	All Dispensing	2012/13	172500.000000	10300.000000	19200.000000	100200.00
18	All Non-Dispensing	2004/05	127009.028751	10092.089308	11785.240311	80479.080
19	All Non-Dispensing	2005/06	137150.437733	9739.897090	13766.429332	89417.305
20	All Non-Dispensing	2006/07	131868.606894	9111.371366	15145.239396	90186.077
21	All Non-Dispensing	2007/08	NaN	NaN	NaN	NaN
22	All Non-Dispensing	2008/09	NaN	NaN	NaN	NaN
23	All Non-Dispensing	2009/10	149200.000000	10100.000000	19600.000000	100200.00
24	All Non-Dispensing	2010/11	158500.000000	10200.000000	22600.000000	104700.00
25	All Non-Dispensing	2011/12	161300.000000	10200.000000	23800.000000	103700.00
26	All Non-Dispensing	2012/13	165800.000000	10900.000000	26700.000000	111300.00

7a. GPMS by NHS ER: 4 tables, 6 columns

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	All Practice Types	2011/12	7950	265900	161900	103900
1	All Practice Types	2012/13	8000	270200	167700	102500
2	All Dispensing	2011/12	1000	329000	216400	112500
3	All Dispensing	2012/13	1000	325600	217700	107800
4	All Non-	2011/12	6950	256800	154100	102700

	Dispensing					
5	All Non-Dispensing	2012/13	6950	262100	160500	101700

7b. GPMS by NHS ER: 4 tables, 10 columns

	Practice Type	Year	Total Expenses1	Office & General Business2	Premises	Employee	Car and Travel	Int
0	All Practice Types	2011/12	161900	15000	21600	99600	1200	190
1	All Practice Types	2012/13	167700	15600	23300	103700	1200	190
2	All Dispensing	2011/12	216400	22300	18400	104000	1600	350
3	All Dispensing	2012/13	217700	22700	19100	109700	1600	310
4	All Non-Dispensing	2011/12	154100	13900	22000	99000	1200	160
5	All Non-Dispensing	2012/13	160500	14600	24000	102900	1100	170

8. GPMS Distribution: 0 tables, 1 columns

	NO TABLES

9a. All Salaried: 4 tables, 6 columns

	Contract Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before T
0	All Salaried: Assistant/Salaried Performer GP...	2002/032	1200	NaN	NaN	43605.00
1	All Salaried: Assistant/Salaried Performer GP...	2003/042	1747	NaN	NaN	47069.00
2	All Salaried: Assistant/Salaried Performer GP...	2004/05 (on majority rule)3	2009	50665.00000	5105.000000	45560.00

3	All Salaried: Assistant/Salaried Performer GP...	2005/06 (on majority rule)3	2743	52172.51261	5267.471666	46905.04
4	All Salaried: Assistant/Salaried Performer GP...	2006/074	NaN	NaN	NaN	NaN
5	All Salaried: Assistant/Salaried Performer GP...	2007/08	NaN	NaN	NaN	NaN
6	All Salaried: Assistant/Salaried Performer GP...	2008/09	NaN	NaN	NaN	NaN
7	All Salaried: Assistant/Salaried Performer GP...	2009/10	NaN	NaN	NaN	NaN
8	All Salaried: Assistant/Salaried Performer GP...	2010/11	NaN	NaN	NaN	NaN
9	All Salaried: Assistant/Salaried Performer GP...	2011/12	NaN	NaN	NaN	NaN
10	All Salaried: Assistant/Salaried Performer GP...	2012/13	NaN	NaN	NaN	NaN

9b. Salaried by Country : 4 tables, 8 columns

	Contract Type	Year	Estimated Population	Gross Employment Earnings	Gross Self Employment Earnings	Total Gross Earnings	
0	GPMS	2006/07	5069.00	47108.093622	13028.438098	60136.531720	
1	GPMS	2007/08	5112.00	49759.899773	12257.248527	62017.148300	
2	GPMS	2008/09	6507.00	50300.000000	13700.000000	64000.000000	
3	GPMS	2009/10	7150.00	50500.000000	14500.000000	65000.000000	
4	GPMS	2010/11	7550.00	49800.000000	14900.000000	64700.000000	
5	GPMS	2011/12	7650.00	49300.000000	14600.000000	63900.000000	
6	GPMS	2012/13	8200.00	49000.000000	15600.000000	64600.000000	
7	GMS	2006/07	2647.58	45580.833814	13327.180716	58908.014534	
8	GMS	2007/08	2002.34	47098.363415	13420.702843	60519.066257	

9	GMS	2008/09	2662.48	48000.000000	15200.000000	63200.000000	
10	GMS	2009/10	3000.00	48600.000000	14400.000000	63100.000000	
11	GMS	2010/11	3250.00	48100.000000	15600.000000	63600.000000	
12	GMS	2011/12	3450.00	46800.000000	15000.000000	61900.000000	
13	GMS	2012/13	3600.00	46800.000000	15900.000000	62600.000000	
14	PMS	2006/07	2421.42	48777.999100	12701.792993	61479.792093	
15	PMS	2007/08	3109.66	51473.688648	11508.089093	62981.777744	
16	PMS	2008/09	3844.52	51800.000000	12800.000000	64600.000000	
17	PMS	2009/10	4150.00	51900.000000	14600.000000	66500.000000	
18	PMS	2010/11	4300.00	51100.000000	14500.000000	65600.000000	
19	PMS	2011/12	4200.00	51300.000000	14200.000000	65500.000000	
20	PMS	2012/13	4600.00	50700.000000	15300.000000	66100.000000	

9c. Salaried by Age: 4 tables, 8 columns

	Contract Type	Year	Estimated Population	Gross Employment Earnings	Gross Self Employment Earnings	Total Gross Earnings	T E
0	GPMS	2006/07	2899.61	47587.819848	12055.962412	59643.782257	5
1	GPMS	2007/08	2897.15	50103.136914	11959.647474	62062.784388	6
2	GPMS	2008/09	3771.89	50000.000000	12100.000000	62200.000000	6
3	GPMS	2009/10	4100.00	50300.000000	13500.000000	63800.000000	7
4	GPMS	2010/11	4350.00	49500.000000	14300.000000	63900.000000	7
5	GPMS	2011/12	4400.00	49000.000000	14500.000000	63500.000000	7
6	GPMS	2012/13	4650.00	48700.000000	15000.000000	63700.000000	8

9d. Salaried by Rurality: 2 tables, 8 columns

	Contract Type	Year	Estimated Population	Gross Employment Earnings	Gross Self Employment Earnings	Total Gross Earnings	T E
0	GPMS	2006/07	957.35	44977.667363	13283.562260	58261.229634	6
1	GPMS	2007/08	932.22	46828.955590	11599.950602	58428.906181	5
2	GPMS	2008/09	1166.17	48400.000000	13800.000000	62100.000000	7
3	GPMS	2009/10	1300.00	48500.000000	13000.000000	61500.000000	6
4	GPMS	2010/11	1350.00	48500.000000	15500.000000	64000.000000	7
5	GPMS	2011/12	1350.00	48000.000000	14200.000000	62200.000000	7

6	GPMS	2012/13	1500.00	47700.000000	15700.000000	63400.000000	8

9e. Salaried by SHA_GOR: 10 tables, 8 columns

	Contract Type	Year	Estimated Population	Gross Employment Earnings	Gross Self Employment Earnings	Total Gross Earnings	T E
0	GPMS	2006/07	212.22	53676.587174	10819.270097	64495.857271	4
1	GPMS	2007/08	232.87	53684.827114	10090.645424	63775.472538	4
2	GPMS	2008/09	362.73	53500.000000	11500.000000	65000.000000	5
3	GPMS	2009/10	350.00	52600.000000	13700.000000	66300.000000	6
4	GPMS	2010/11	400.00	51900.000000	15000.000000	66900.000000	6
5	GPMS	2011/12	400.00	51100.000000	13200.000000	64200.000000	5
6	GPMS	2012/13	400.00	50500.000000	14800.000000	65300.000000	6

9f. Salaried by NHS ER: 4 tables, 8 columns

	Contract Type	Year	Estimated Population	Gross Employment Earnings	Gross Self Employment Earnings	Total Gross Earnings	Total Expense
0	GPMS	2011/12	1550	51400	14500	65900	6800
1	GPMS	2012/13	1700	50300	15900	66200	7900

10. Salaried Distribution : 0 tables, 1 columns

	NO TABLES

11. Combined GPs: 1 tables, 6 columns

	Contract Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Tax
0	GPMS	2006/07	38956.00	222999.653460	122323.463409	100676.1900
1	GPMS	2007/08	38734.00	226924.255019	127488.464246	99435.79077
2	GPMS	2008/09	39878.00	226900.000000	129400.000000	97500.00000
3	GPMS	2009/10	40200.00	227500.000000	130300.000000	97200.00000
4	GPMS	2010/11	40550.00	228900.000000	133400.000000	95400.00000
5	GPMS	2011/12	40600.00	229400.000000	135200.000000	94200.00000
6	GPMS	2012/13	41050.00	230400.000000	137500.000000	92900.00000

7	GMS	2006/07	26604.00	215674.834094	117217.913894	98456.92020(
8	GMS	2007/08	23342.34	216855.553655	120515.475031	96340.07862
9	GMS	2008/09	24031.49	218000.000000	123700.000000	94300.00000(
10	GMS	2009/10	24250.00	219200.000000	124400.000000	94900.00000(
11	GMS	2010/11	24600.00	220500.000000	127200.000000	93300.00000(
12	GMS	2011/12	24900.00	221200.000000	128900.000000	92300.00000(
13	GMS	2012/13	25000.00	222700.000000	131600.000000	91100.00000(
14	PMS	2006/07	12352.00	241351.526150	135209.951893	106141.5742
15	PMS	2007/08	15391.66	242194.021282	138063.404511	104130.6167
16	PMS	2008/09	15846.51	240200.000000	138000.000000	102200.0000(
17	PMS	2009/10	15950.00	240100.000000	139300.000000	100800.0000(
18	PMS	2010/11	15950.00	241800.000000	143000.000000	98800.00000(

This is where we check reading the worksheets in `pandas`. This time we are going to do as much of the work as we can on dataframes. A lot of this work was actually used to determine how to write the `extract_tables()` function above, so it's really out of order. Sometimes life just is that way - messy and refusing to fit into the nice neat patterns we carry around in our head.

Basically I just refactored the code below making the worksheet a parameter rather than a (module) global variable.

In [71]:

```
from pandas.io.excel import read_excel
pwb = read_excel("data/gpearnextime.xls", sheetname=None)
```

In [72]:

```
psheet = pwb['1a. GPMS Cash Terms ']
```

In [73]:

```
psheet.index = range(len(psheet)) # This may no longer be necessary
```

In [74]:

```
# Occasional random cells have odd characters in them
for name in list(psheet.columns.values):
     if psheet[name].count() <= 1:
        del psheet[name]
```

In [75]:

```
# Establish the limits of the tables
row_counts = list(psheet.count(axis=1).values)
endpos = len(row_counts)-1
while row_counts[endpos] < 2:
    endpos -= 1

startpos = 0
while startpos < endpos-1 and (row_counts[startpos] != 1 or row_counts[star
tpos+1] <= 1):
    startpos += 1
```

In [76]:

```
# We assume there is no table of three columns or less
row_counts = list(psheet.count(axis=1).values)
start_row_nums = []
for (i, val) in enumerate(row_counts[:-1]):
    if val==1:
        if row_counts[i+1] != row_counts[i+2]:
            continue
        else:
            if row_counts[i+1] > 3:
                start_row_nums.append(i)
```

In [77]:

```
tables = []
titles = []
extend = True
for start_row_num in start_row_nums:
    columns = list(psheet.iloc[start_row_num+1,:])
    titles.append(psheet.iloc[start_row_num, 0])
    end_row_num = start_row_num+1
    while row_counts[end_row_num] > 0:
        end_row_num += 1
    table = pd.DataFrame(psheet.iloc[start_row_num+2:end_row_num,:])
    table.index = range(end_row_num-start_row_num-2) # No title, no headers
    table.columns = columns
    table[headers[0]] = table[headers[0]].fillna(method="ffill")
    table.replace("-", np.NaN, inplace=True)
    if extend:
        table["Geography"] = titles[-1]
    tables.append(table)
```

In [78]:

```
btbl_len = 0
btbl_len = sum(len(table) for table in tables)
big_table = pd.concat(tables)
big_table.index = range(btbl_len)
```

The rest of the notebook is essentially me demonstrating my ineptitude with `pandas`.

In [79]:

```
big_table
```

Out[79]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Incom Before
0	All Practice Types	2002/031	31362	184154.000000	111439.000000	72716.
1	All Practice Types	2003/041	31594.98	203613.465384	121594.750565	82018.
2	All Practice Types	2004/05	33887.99	230096.703402	129926.179861	10016
3	All Practice Types	2005/06	33874.98	245019.575577	135015.975718	11000
4	All Practice Types	2006/07	33887	247361.596509	139694.391103	10766
5	All Practice Types	2007/08	33622	251997.276539	145925.375805	10607
6	All Practice Types	2008/09	33371	258600.000000	153300.000000	10530
7	All Practice Types	2009/10	33050	262700.000000	156900.000000	10570
8	All Practice Types	2010/11	33000	266500.000000	162400.000000	10410
9	All Practice Types	2011/12	32950	267900.000000	164900.000000	10300
10	All Practice Types	2012/13	32850	271800.000000	169700.000000	10200
11	All Dispensing	2002/031	4851	271003.000000	183830.000000	87172.
[illegible]	All	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

12	Dispensing	2003/041	5233.81	286133.318928	188593.127027	97540.
13	All Dispensing	2004/05	5329.23	317973.891450	198418.141182	11955
14	All Dispensing	2005/06	5303.1	331894.907881	204833.866238	12706

In [80]:

```
big_table.set_index(["Geography","Practice Type"])
```

Out[80]:

		Year	Estimated Population	Gross Earnings	Total Expenses
Geography	**Practice Type**				
	All Practice Types	2002/031	31362	184154.000000	111439.00000
	All Practice Types	2003/041	31594.98	203613.465384	121594.75056
	All Practice Types	2004/05	33887.99	230096.703402	129926.17986
	All Practice Types	2005/06	33874.98	245019.575577	135015.97571
	All Practice Types	2006/07	33887	247361.596509	139694.39110
	All Practice Types	2007/08	33622	251997.276539	145925.37580
	All Practice Types	2008/09	33371	258600.000000	153300.00000
	All Practice Types	2009/10	33050	262700.000000	156900.00000
	All Practice Types	2010/11	33000	266500.000000	162400.00000
	All				

GPMS UK	**Practice Types**	2011/12	32950	267900.000000	164900.00000
	All Practice Types	2012/13	32850	271800.000000	169700.00000
	All Dispensing	2002/031	4851	271003.000000	183830.00000
	All Dispensing	2003/041	5233.81	286133.318928	188593.12702
	All Dispensing	2004/05	5329.23	317973.891450	198418.14118

In [81]:

```
pt_geo = big_table.set_index(["Geography"])
```

In [82]:

```
pt_geo.index.values
```

Out[82]:

```
array(['GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK',
'GPMS UK',
       'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK',
'GPMS UK',
       'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK',
'GPMS UK',
       'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK',
'GPMS UK',
       'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS UK',
'GPMS UK',
       'GPMS UK', 'GPMS UK', 'GPMS UK', 'GPMS ENGLAND', 'GPMS E
NGLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS EN
GLAND',
       'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS ENGLAND', 'GPMS SC
OTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
```

```
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS

SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS SCOTLAND', 'GPMS
SCOTLAND',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS WALES', 'GPMS WALES', 'GPMS WALES',
       'GPMS WALES', 'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IR
ELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND', 'GPMS NORTHERN IRELAND',
       'GPMS NORTHERN IRELAND'], dtype=object)
```

In [83]:

```
pt_geo[pt_geo.index.values == 'GPMS NORTHERN IRELAND']
```

Out[83]:

	Practice Type	**Year**	**Estimated Population**	**Gross Earnings**	**Total Expenses**
Geography					
GPMS NORTHERN IRELAND	All Practice Types	2002/03	1067	128927.000000	68127.000000
GPMS NORTHERN IRELAND	All Practice Types	2003/04	1021	140243.169334	72679.445779

GPMS NORTHERN IRELAND	All Practice Types	2004/05	1045	173091.264561	81940.070172
GPMS NORTHERN IRELAND	All Practice Types	2005/06	1053	185204.976572	86548.770940
GPMS NORTHERN IRELAND	All Practice Types	2006/07	1108	181892.341164	88576.701661
GPMS NORTHERN IRELAND	All Practice Types	2007/08	1123	181029.451941	89973.789911
GPMS NORTHERN IRELAND	All Practice Types	2008/09	1144	183700.000000	94000.000000
GPMS NORTHERN IRELAND	All Practice Types	2009/10	1100	189200.000000	97800.000000
GPMS NORTHERN IRELAND	All Practice Types	2010/11	1150	185700.000000	97700.000000
GPMS NORTHERN IRELAND	All Practice Types	2011/12	1150	192600.000000	99900.000000
GPMS NORTHERN IRELAND	All Practice Types	2012/13	1150	191100.000000	99000.000000
GPMS NORTHERN IRELAND	All Dispensing	2002/03	26	246983.000000	165307.000000
GPMS NORTHERN IRELAND	All Dispensing	2003/04	24	273427.646746	186045.806865
GPMS					

This allows us to select the subset of the data referring to a specific geography. In this particular case we can see that the "Estimated Population" column has some unusable values in it (which have caused it to be interpreted as a string column). Also the financials columns for some of the practice types are absent for some years.

In [84]:

```
big_table[big_table.Geography == 'GPMS NORTHERN IRELAND']
```

Out[84]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before T
132	All Practice Types	2002/03	1067	128927.000000	68127.000000	60800.0
133	All Practice Types	2003/04	1021	140243.169334	72679.445779	67563.7
134	All Practice Types	2004/05	1045	173091.264561	81940.070172	91151.1
135	All Practice Types	2005/06	1053	185204.976572	86548.770940	98656.2
136	All Practice Types	2006/07	1108	181892.341164	88576.701661	93315.6
137	All Practice Types	2007/08	1123	181029.451941	89973.789911	91055.6
138	All Practice Types	2008/09	1144	183700.000000	94000.000000	89700.0
139	All Practice Types	2009/10	1100	189200.000000	97800.000000	91400.0
140	All Practice Types	2010/11	1150	185700.000000	97700.000000	88000.0
141	All Practice Types	2011/12	1150	192600.000000	99900.000000	92800.0
142	All Practice Types	2012/13	1150	191100.000000	99000.000000	92200.0
143	All Dispensing	2002/03	26	246983.000000	165307.000000	81677.0
144	All Dispensing	2003/04	24	273427.646746	186045.806865	87381.8

145	All Dispensing	2004/05	19	NaN	NaN	NaN

Next we want to investigate the grouping, to allow us to summarize the data as we want.

In [85]:

```
big_table.groupby("Geography")
```

Out[85]:

```
<pandas.core.groupby.DataFrameGroupBy object at 0x114d58550>
```

In [86]:

```
for x in big_table.groupby(["Geography", "Practice Type"]):
    t, df = x
    break
print(t)
```

```
('GPMS ENGLAND', 'All Dispensing')
```

In [87]:

```
df
```

Out[87]:

	Practice Type	Year	Estimated Population	Gross Earnings	Total Expenses	Income Before Ta
44	All Dispensing	2002/03	4281	274864.000000	186620.000000	88244.00
45	All Dispensing	2003/04	4621.276	290995.531436	192121.770137	98873.76
46	All Dispensing	2004/053	4695.8	NaN	202306.786358	NaN
47	All Dispensing	2005/06	4637.35	340051.668298	210298.989183	129752.6
48	All Dispensing	2006/07	4668	337134.012421	207777.110482	129356.9
49	All Dispensing	2007/08	4528	345118.812137	217854.639523	127264.1
50	All Dispensing	2008/09	4302	354800.000000	231000.000000	123800.0
51	All Dispensing	2009/10	4250	356600.000000	232900.000000	123600.0
52	All Dispensing	2010/11	4400	365800.000000	245200.000000	120500.0
53	All Dispensing	2011/12	4400	367000.000000	249500.000000	117500.0
54	All Dispensing	2012/13	4350	369900.000000	254000.000000	115900.0

In [88]:

```
df.plot("Year")
```

Out[88]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x114d5e128>
```

In [89]:

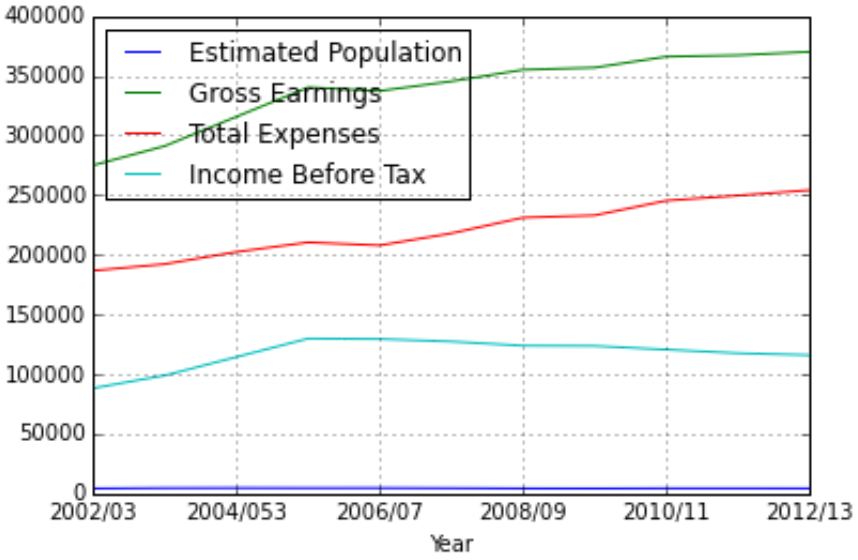
```
df.interpolate("linear").plot("Year")
```

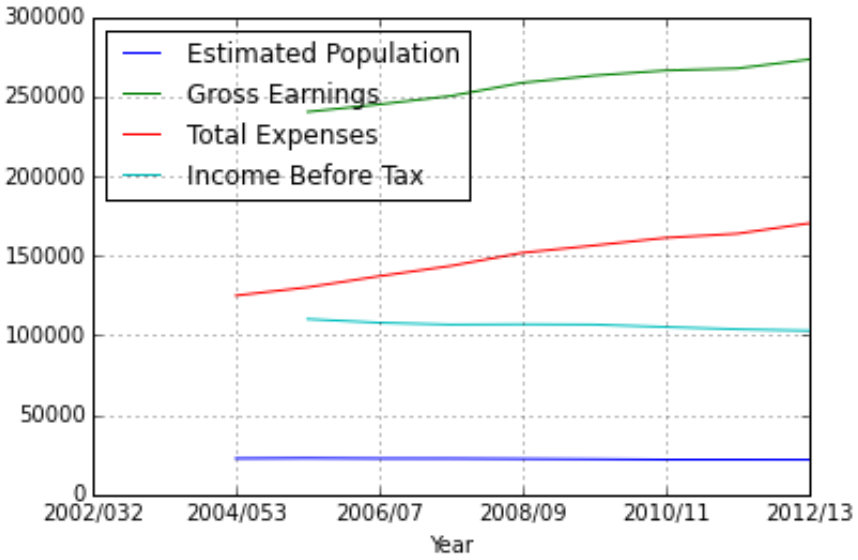
Out[89]:

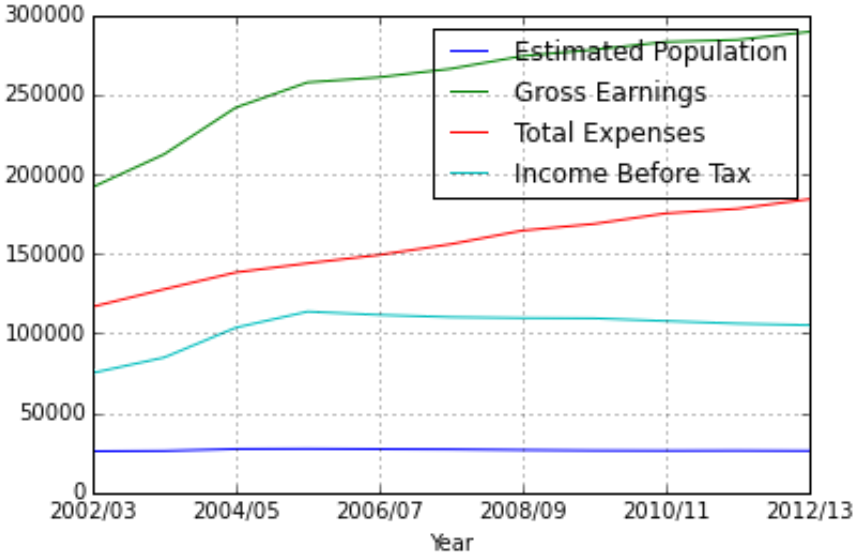
```
<matplotlib.axes._subplots.AxesSubplot at 0x114e40ac8>
```

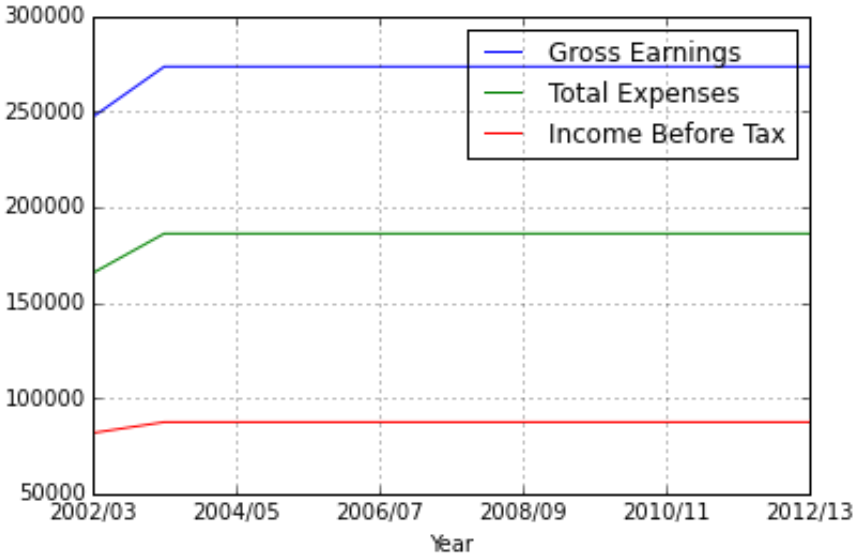
In [90]:

```
for x in big_table.groupby(["Geography", "Practice Type"]):
    t, df = x
    df.interpolate("linear").plot("Year")
```

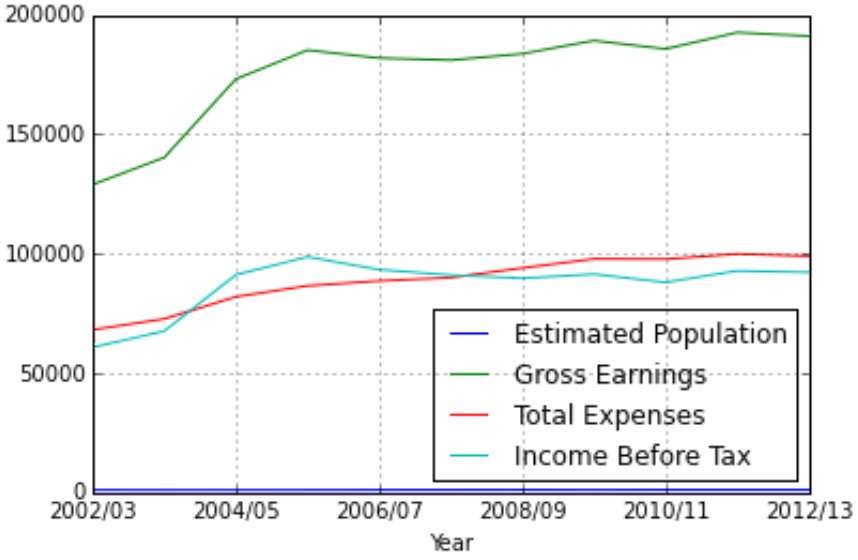

Estimated Population
Gross Earnings
Total Expenses
Income Before Tax
400000
350000
300000
250000
200000
150000
100000
50000
0
2002/03
2004/053
2006/07
2008/09
2010/11
2012/13
Year

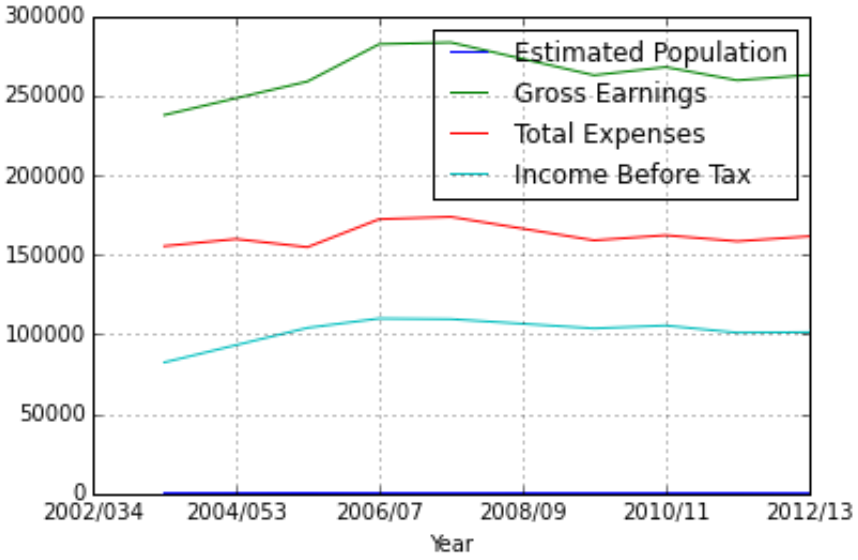

Estimated Population
Gross Earnings
Total Expenses
Income Before Tax
300000
250000
200000
150000
100000
50000
0
2002/032
2004/053
2006/07
2008/09
2010/11
2012/13
Year

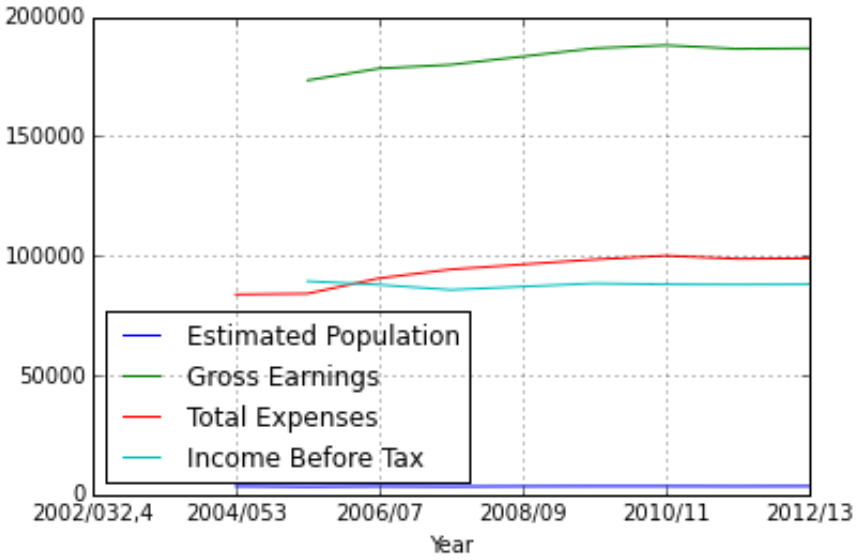

Estimated Population
Gross Earnings
Total Expenses
Income Before Tax
300000
250000
200000
150000
100000
50000
0
2002/03
2004/05
2006/07
2008/09
2010/11
2012/13
Year

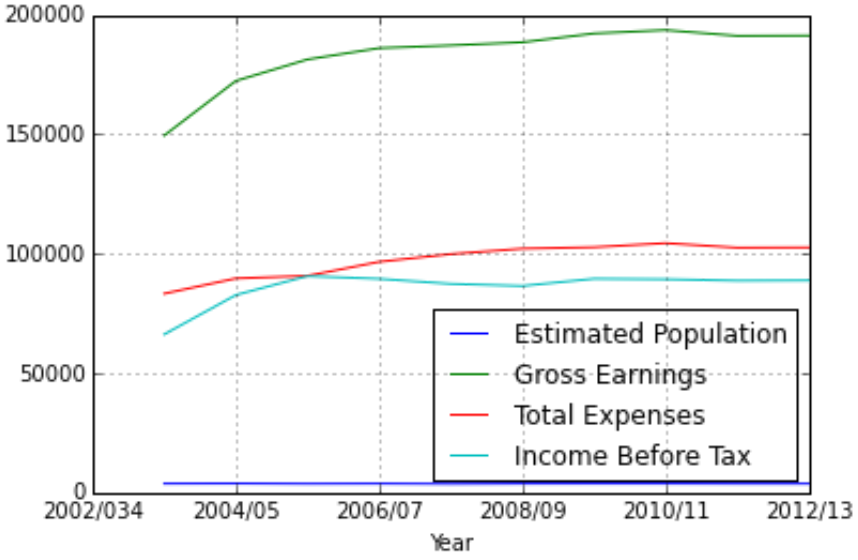

300000
250000
200000
150000
100000
50000
2002/03
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Gross Earnings
Total Expenses
Income Before Tax

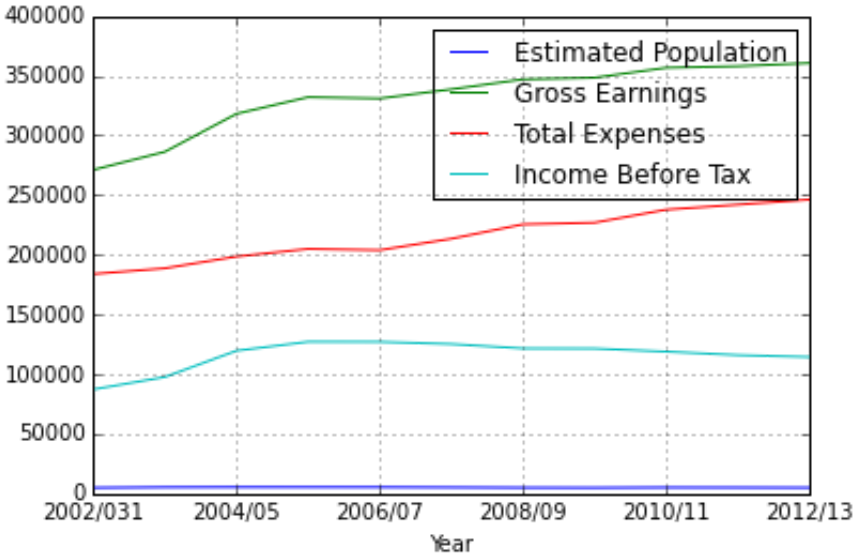

200000
150000
100000
50000
0
2002/032
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax

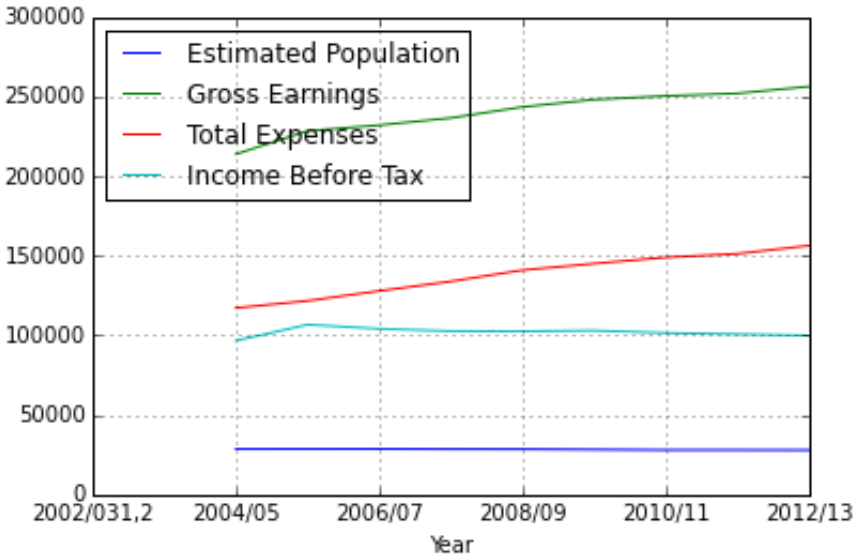

200000
150000
100000
50000
0
2002/03
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax

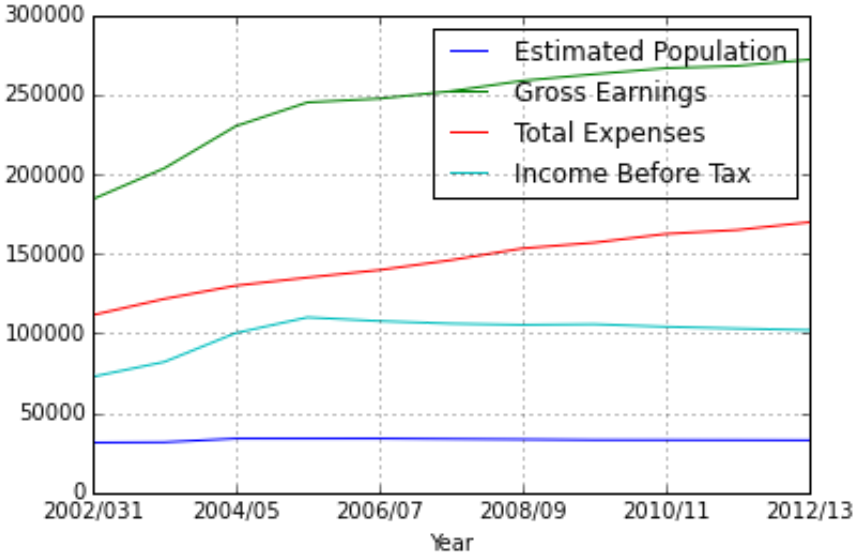

300000
250000
200000
150000
100000
50000
0
2002/034
2004/053
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax

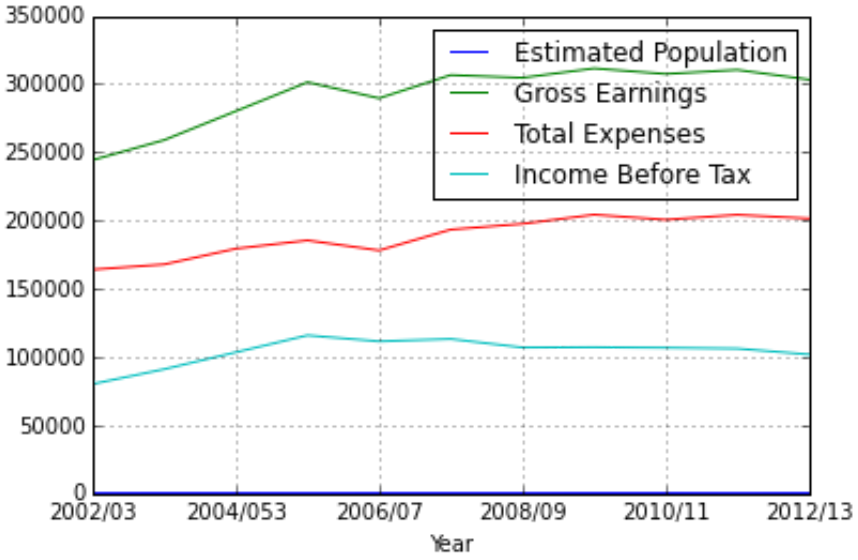

200000
150000
100000
50000
0
2002/032,4
2004/053
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax

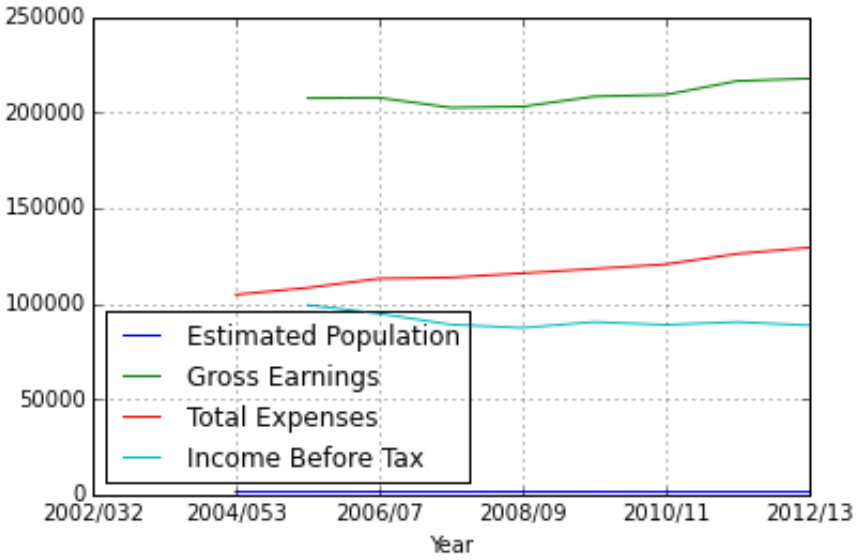

200000
150000
100000
50000
0
2002/034
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax

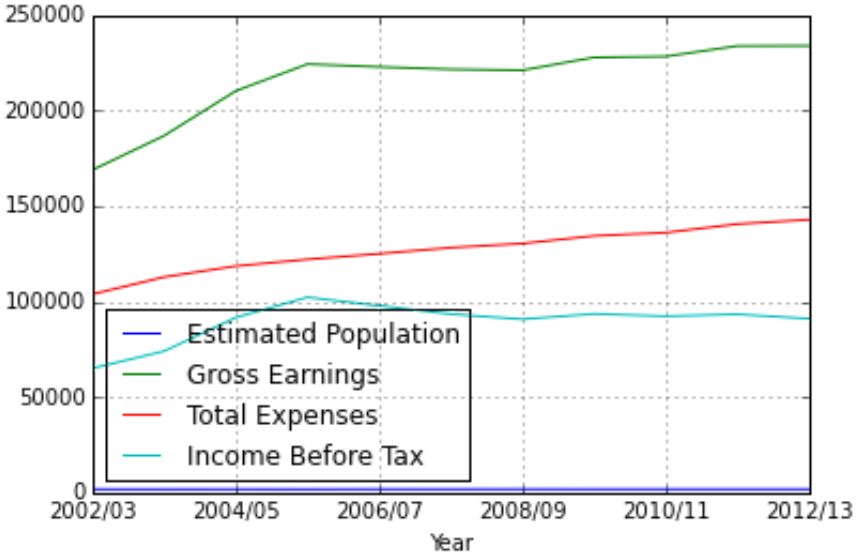

400000
350000
300000
250000
200000
150000
100000
50000
0
2002/031
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax


300000
250000
200000
150000
100000
50000
0
2002/031,2
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax


300000
250000
200000
150000
100000
50000
0
2002/031
2004/05
2006/07
2008/09
2010/11
2012/13
Year
Estimated Population
Gross Earnings
Total Expenses
Income Before Tax







In [91]:

```
for x in big_table.groupby(["Geography", "Practice Type"]):
    t, df = x
    print(df.Geography.iloc[0], df[["Practice Type"]].iloc[0], '\n', df.int
erpolate("linear").describe())
```

GPMS ENGLAND Practice Type All Dispensing
Name: 44, dtype: object

	Gross Earnings	Total Expenses	Income Before Tax
count	11.000000	11.000000	11.000000
mean	337980.693105	220870.845062	117191.339561
std	31648.137050	23222.702335	12986.420683
min	274864.000000	186620.000000	88244.000000
25%	326328.806144	205041.948420	115106.610104
50%	345118.812137	217854.639523	120500.000000
75%	361200.000000	239050.000000	125532.086307
max	369900.000000	254000.000000	129752.679116

GPMS ENGLAND Practice Type All Non-Dispensing
Name: 55, dtype: object

	Gross Earnings	Total Expenses	Income Before Tax
count	8.000000	9.000000	8.000000
mean	258047.864926	148804.889530	106284.892566
std	11711.481412	15770.762942	2307.909643
min	240310.005865	124940.226894	102900.000000
25%	248955.924349	137103.881392	104850.000000
50%	260850.000000	151800.000000	106703.582710
75%	266625.000000	161200.000000	107148.410057
max	273200.000000	170300.000000	110178.334882

GPMS ENGLAND Practice Type All Practice Types
Name: 33, dtype: object

	Gross Earnings	Total Expenses	Income Before Tax
count	11.000000	11.000000	11.000000
mean	258116.085000	154781.163691	103334.921310
std	31231.990864	21758.212088	12108.786784
min	191777.000000	116671.000000	75106.000000
25%	249679.422046	141090.516788	104332.119962
50%	266110.022166	155970.751418	107700.000000
75%	280550.000000	172000.000000	109869.635374
max	289300.000000	184200.000000	113613.570591

GPMS NORTHERN IRELAND Practice Type All Dispensing
Name: 143, dtype: object

	Gross Earnings	Total Expenses	Income Before Tax
count	11.000000	11.000000	11.000000
mean	271023.587951	184160.460786	86863.218074
std	7973.360997	6252.985543	1720.073943
min	246983.000000	165307.000000	81677.000000
25%	273427.646746	186045.806865	87381.839881
50%	273427.646746	186045.806865	87381.839881
75%	273427.646746	186045.806865	87381.839881
max	273427.646746	186045.806865	87381.839881

GPMS NORTHERN IRELAND Practice Type All Non-Dispensing
Name: 154, dtype: object

	Gross Earnings	Total Expenses	Income Before Tax
count	11.000000	11.000000	11.000000
mean	166250.721191	85679.693783	80571.027408
std	23057.563969	12110.619059	10988.710809
min	125978.000000	65700.000000	60278.000000
25%	149602.842807	76537.849632	73064.993175
50%	170545.707914	87516.822059	83028.885855
75%	185950.000000	96550.000000	89400.000000
max	192300.000000	99600.000000	92700.000000

GPMS NORTHERN IRELAND Practice Type All Practice Types
Name: 132, dtype: object

```
        Gross Earnings  Total Expenses  Income Before Tax
count       11.000000       11.000000          11.000000
mean    175698.927597    88749.616224       86967.493192
std      21160.856751    10748.356098       11670.314734
min     128927.000000    68127.000000       60800.000000
25%     177060.358251    84244.420556       88850.000000
50%     183700.000000    89973.789911       91151.194388
75%     187450.000000    97750.000000       92500.000000
max     192600.000000    99900.000000       98656.205632
GPMS SCOTLAND Practice Type    All Dispensing
Name: 77, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       10.000000       10.000000          10.000000
mean    263596.703547   162340.674444      101714.633903
std      14092.412832     6546.104328        8332.933194
min     237704.695518   154714.137801       82337.708829
25%     258945.681434   158550.000000      101075.000000
50%     262750.000000   160663.305123      103873.385389
75%     271695.325090   165329.824836      106340.500253
max     283287.533573   173746.199564      109907.618604
GPMS SCOTLAND Practice Type    All Non-Dispensing
Name: 88, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count        8.000000        9.000000           8.000000
mean    182922.510381    93884.067021       87771.567438
std       5222.750419     6342.415031        1046.226906
min     173398.995406    83749.059639       85697.221693
25%     179549.210770    90532.363254       87636.879425
50%     184985.959240    96323.307633       87950.000000
75%     186800.000000    98700.000000       88100.000000
max     188100.000000   100000.000000       89293.738010
GPMS SCOTLAND Practice Type    All Practice Types
Name: 66, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       10.000000       10.000000          10.000000
mean    183318.005280    97432.275690       85915.729590
std      13458.225520     7177.113907        7275.986901
min     149469.898894    83265.298594       66204.600300
25%     182506.174068    92175.334923       86717.680978
50%     187848.514078   100963.152122       88750.000000
75%     191275.000000   102575.000000       89425.767089
max     193600.000000   104400.000000       90618.555709
GPMS UK Practice Type    All Dispensing
Name: 11, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       11.000000       11.000000          11.000000
mean    331454.080571   215518.533645      115944.546926
std      29352.136241    21508.346521       12577.103017
min     271003.000000   183830.000000       87172.000000
25%     324382.366151   201106.341813      115100.000000
50%     338498.927168   213334.193203      119555.750268
75%     352350.000000   232250.000000      123332.366983
max     360400.000000   246100.000000      127061.041643
GPMS UK Practice Type    All Non-Dispensing
Name: 22, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count        9.000000        9.000000           9.000000
```

```
mean    240027.548052   138082.197165      101956.461998
std      13644.657947    13732.287737        2817.750272
min     213738.267866   117145.183241       96593.084625
25%     231935.398247   127842.172485      100700.000000
50%     243400.000000   140900.000000      102500.000000
75%     250300.000000   148900.000000      103000.000000
max     256200.000000   156300.000000      106680.507156
GPMS UK Practice Type    All Practice Types
Name: 0, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       11.000000       11.000000          11.000000
mean    244522.056128   144617.788459       99886.085851
std      28093.354872    18835.620724       11635.056071
min     184154.000000   111439.000000       72716.000000
25%     237558.139490   132471.077789      101084.761771
50%     251997.276539   145925.375805      104100.000000
75%     264600.000000   159650.000000      105885.950367
max     271800.000000   169700.000000      110003.599860
GPMS WALES Practice Type    All Dispensing
Name: 110, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       11.000000       11.000000          11.000000
mean    292569.326472   188777.812272      104048.293771
std      22467.635512    14621.283948       10246.746305
min     244332.000000   164088.000000       80245.000000
25%     284891.961174   178763.544355      102696.273689
50%     303300.000000   193390.225117      106800.000000
75%     307005.476459   201050.000000      109343.930769
max     311500.000000   204300.000000      115891.968986
GPMS WALES Practice Type    All Non-Dispensing
Name: 121, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count        8.000000        9.000000           8.000000
mean    209022.870802   116551.276824       90991.067693
std       5523.318610     7919.187185        3996.452120
min     202495.852993   104607.066552       87200.000000
25%     206320.297447   112978.721260       88725.000000
50%     207980.025080   115800.000000       89616.643852
75%     211000.000000   120500.000000       91395.332225
max     217700.000000   129200.000000       99213.924943
GPMS WALES Practice Type    All Practice Types
Name: 99, dtype: object
        Gross Earnings  Total Expenses  Income Before Tax
count       11.000000       11.000000          11.000000
mean    216237.842952   126736.658206       89518.943901
std      20484.443841    11908.009253       10637.249027
min     168842.000000   103836.000000       65007.000000
25%     215592.172628   120253.696506       90850.000000
50%     222729.232702   128130.319011       92300.000000
75%     227950.000000   135150.000000       93432.751944
max     233800.000000   142800.000000      102193.888055
```

from collections import defaultdict g = {} for t, df in big_table.groupby(["Geography", "Practice Type"]): df = df.interpolate("linear") g[df.Geography.iloc[0], df[["Practice Type"]].iloc[0]] = \ df[["Income Before Tax"]].mean() print(g)

In [93]:

```
pd.options.display.float_format = '£{:,.0f}'.format

ads = big_table.groupby("Geography").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
```

```
                       Total Expenses  Gross Earnings  Income B
efore Tax
Geography
GPMS ENGLAND                 £176,497        £286,411
£109,026
GPMS NORTHERN IRELAND         £97,603        £182,685
£85,093
GPMS SCOTLAND                £117,777        £209,117
£91,665
GPMS UK                      £167,879        £274,064
£106,185
GPMS WALES                   £145,794        £240,996
£94,954
```

In [94]:

```
pd.options.display.float_format = '{}'.format
ads[abs(ads["Income Before Tax"]+ads["Total Expenses"]-ads["Gross Earning
s"])>500]
```

Out[94]:

	Total Expenses	Gross Earnings	Income Before Tax
Geography			
GPMS ENGLAND	176497.29361467133	286411.16823118087	109026.02723799922

In [95]:

```
df[["Practice Type", "Income Before Tax"]]
```

Out[95]:

	Practice Type	Income Before Tax
44	All Dispensing	88244.0
45	All Dispensing	98873.76129868152
46	All Dispensing	114313.2202071474
47	All Dispensing	129752.67911561328
48	All Dispensing	129356.90193823782
49	All Dispensing	127264.17261367393
50	All Dispensing	123800.0
51	All Dispensing	123600.0
52	All Dispensing	120500.0
53	All Dispensing	117500.0
54	All Dispensing	115900.0

In [96]:

```
for t, df in big_table.groupby(["Geography", "Practice Type"]):
    print(df[["Income Before Tax"]].mean())
```

```
Income Before Tax   117479.15149662066
dtype: float64
Income Before Tax   106284.89256620072
dtype: float64
Income Before Tax   103334.92130965138
dtype: float64
Income Before Tax   84529.4199404762
dtype: float64
Income Before Tax   81844.10959441036
dtype: float64
Income Before Tax   86967.49319174794
dtype: float64
Income Before Tax   102166.67902740766
dtype: float64
Income Before Tax   87874.84695061472
dtype: float64
Income Before Tax   85915.72958970108
dtype: float64
Income Before Tax   115944.54692605798
dtype: float64
Income Before Tax   101956.46199803938
dtype: float64
Income Before Tax   99886.0858509855
dtype: float64
Income Before Tax   104103.86840982756
dtype: float64
Income Before Tax   90991.06769333573
dtype: float64
Income Before Tax   89518.94390085731
dtype: float64
```

In [97]:

```
df[["Geography", "Year", "Income Before Tax"]]
```

Out[97]:

	Geography	Year	Income Before Tax
99	GPMS WALES	2002/03	65007.0
100	GPMS WALES	2003/04	73982.78299825684
101	GPMS WALES	2004/05	91587.67622473872
102	GPMS WALES	2005/06	102193.888055053
103	GPMS WALES	2006/07	97771.53174280877
104	GPMS WALES	2007/08	93365.50388857306
105	GPMS WALES	2008/09	90700.0
106	GPMS WALES	2009/10	93500.0
107	GPMS WALES	2010/11	92300.0
108	GPMS WALES	2011/12	93300.0
109	GPMS WALES	2012/13	91000.0

In [98]:

```
pd.options.display.float_format = '£{:,.0f}'.format
ads = big_table.groupby("Practice Type").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
```

```
                  Total Expenses  Gross Earnings  Income Befo
re Tax
Practice Type
All Dispensing            £197,391        £307,842           £1
09,210
All Non-Dispensing        £119,669        £215,651
£94,790
All Practice Types        £122,927        £216,176
£93,258
```

In [99]:

```
ads
```

Out[99]:

	Total Expenses	Gross Earnings	Income Before Tax
Practice Type			
All Dispensing	£197,391	£307,842	£109,210
All Non-Dispensing	£119,669	£215,651	£94,790
All Practice Types	£122,927	£216,176	£93,258

In [100]:

```
pd.options.display.float_format = '£{:,.0f}'.format
ads = big_table.groupby("Year").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
```

```
           Total Expenses  Gross Earnings  Income Before Tax
Year
2002/03          £134,108        £209,288            £75,180
2002/031         £147,634        £227,578            £79,944
2002/031,2           £nan            £nan               £nan
2002/032          £65,700        £125,978            £60,278
2002/032,4           £nan            £nan               £nan
2002/034             £nan            £nan               £nan
2003/04          £137,227        £218,756            £81,529
2003/041         £155,094        £244,873            £89,779
2003/041,2           £nan            £nan               £nan
2003/042          £69,950        £137,037            £67,087
2004/05          £124,827        £222,745            £97,902
2004/053         £142,471            £nan               £nan
2005/06          £136,741        £244,218           £107,477
2006/07          £140,727        £246,268           £105,541
2007/08          £146,411        £250,040           £103,629
2008/09          £155,155        £257,918           £102,791
2009/10          £149,829        £251,600           £101,771
2010/11          £153,493        £253,786           £100,307
2011/12          £155,664        £255,657            £99,993
2012/13          £158,186        £257,021            £98,857
```

In [101]:

```
bt = big_table.copy()
bt["years"] = [year_refs(y)[0] for y in bt["Year"]]
```

In [102]:

```
bt["xrefs"] = [year_refs(y)[1] for y in bt["Year"]]
```

In [103]:

```
pd.options.display.float_format = '£{:,.0f}'.format
ads = bt.groupby("years").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
```

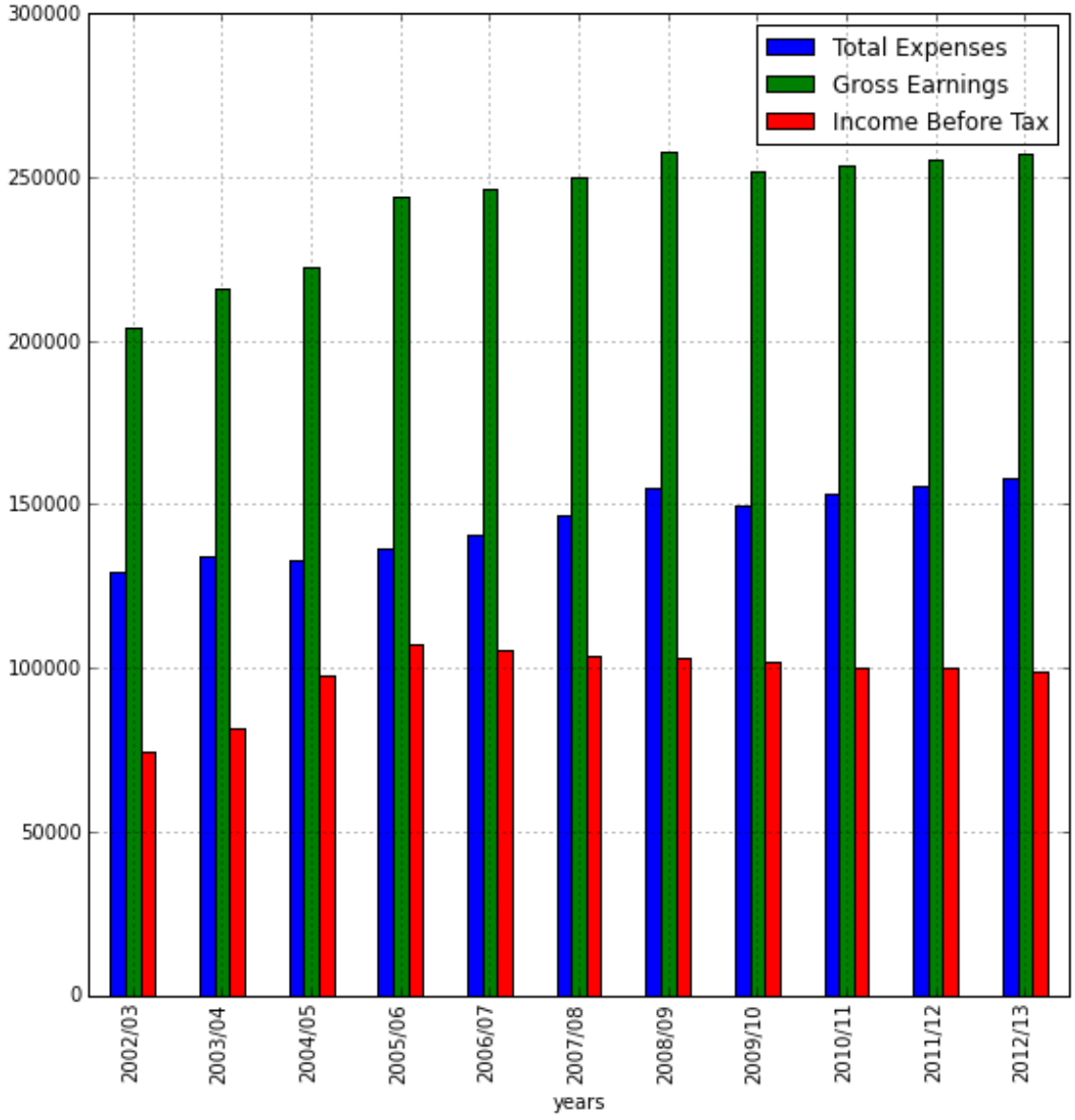
years	Total Expenses	Gross Earnings	Income Before Tax
2002/03	£129,513	£204,096	£74,583
2003/04	£134,359	£216,075	£81,716
2004/05	£132,971	£222,745	£97,902
2005/06	£136,741	£244,218	£107,477
2006/07	£140,727	£246,268	£105,541
2007/08	£146,411	£250,040	£103,629
2008/09	£155,155	£257,918	£102,791
2009/10	£149,829	£251,600	£101,771
2010/11	£153,493	£253,786	£100,307
2011/12	£155,664	£255,657	£99,993
2012/13	£158,186	£257,021	£98,857

In [104]:

```
ads.plot(kind='bar', layout='stack', figsize=(9,9))
```

Out[104]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x115ae4e80>
```



In [105]:

```
bt = bt[bt.Geography != "GPMS UK"]
```

In [106]:

```
pd.options.display.float_format = '£{:,.0f}'.format
ads = bt.groupby("Geography").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
```

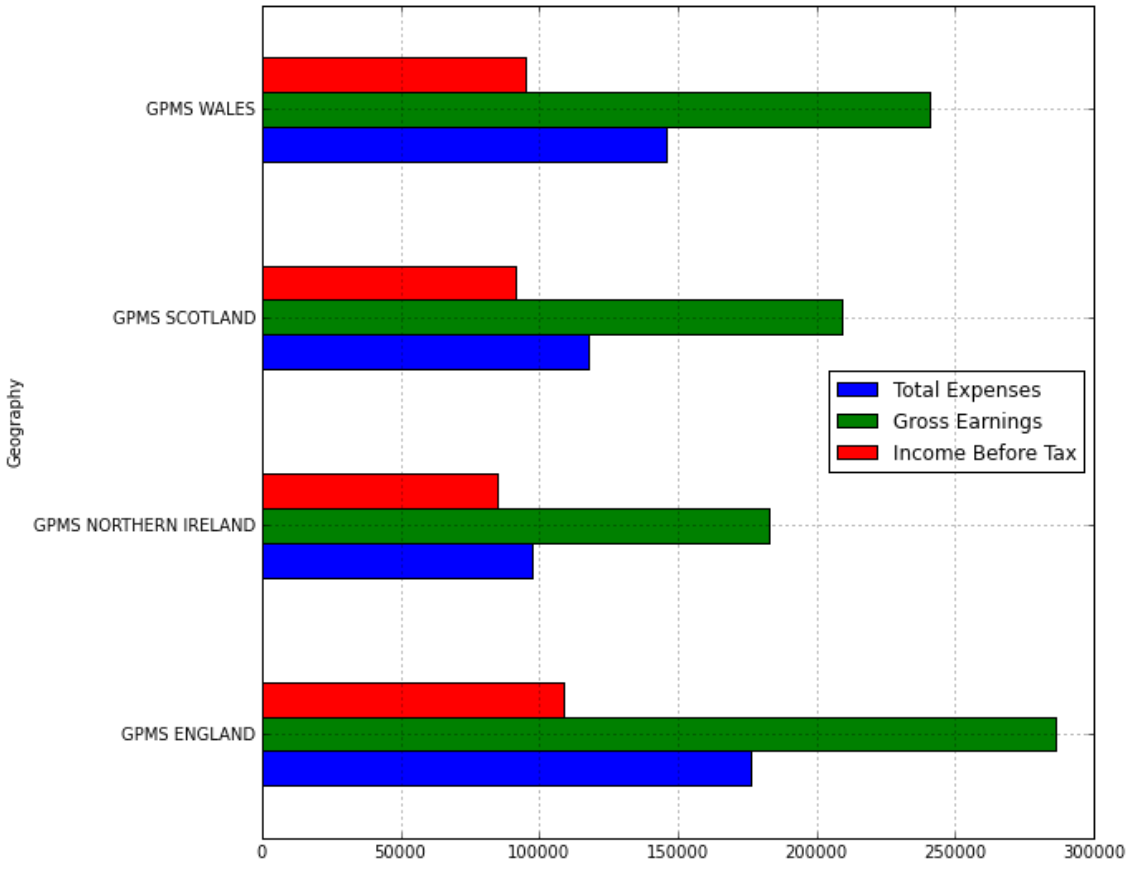
```
                        Total Expenses  Gross Earnings  Income B
efore Tax
Geography
GPMS ENGLAND                  £176,497        £286,411
£109,026
GPMS NORTHERN IRELAND          £97,603        £182,685
£85,093
GPMS SCOTLAND                 £117,777        £209,117
£91,665

GPMS WALES                    £145,794        £240,996
£94,954
```

In [107]:

```
ads.plot(kind='barh', layout='stack', figsize=(9,9))
```

Out[107]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x115a204e0>
```



In [116]:

```
pd.options.display.float_format = '£{:,.0f}'.format
ads = bt.groupby("Year").agg({
        "Gross Earnings": np.mean,
        "Total Expenses": np.mean,
        "Income Before Tax": np.mean
    })
print(ads)
#ads.plot(kind='barh', layout='stack', figsize=(9,9))
```

Year	Total Expenses	Gross Earnings	Income Before Tax
2002/03	£134,108	£209,288	£75,180
2002/032	£65,700	£125,978	£60,278
2002/032,4	£nan	£nan	£nan
2002/034	£nan	£nan	£nan
2003/04	£137,227	£218,756	£81,529
2003/042	£69,950	£137,037	£67,087
2004/05	£107,076	£199,352	£92,250
2004/053	£142,471	£nan	£nan
2005/06	£131,615	£236,960	£105,346
2006/07	£135,812	£239,140	£103,328
2007/08	£141,027	£242,356	£101,330
2008/09	£148,388	£248,538	£100,175
2009/10	£142,627	£242,136	£99,518
2010/11	£145,445	£243,609	£98,173
2011/12	£147,391	£245,600	£98,209
2012/13	£149,318	£246,355	£97,073

In [115]:

```
list((year_refs(dy) for dy in ads.index))
```

Out[115]:

```
[('2002/03', ['']),
 ('2002/03', ['2']),
 ('2002/03', ['2', '4']),
 ('2002/03', ['4']),
 ('2003/04', ['']),
 ('2003/04', ['2']),
 ('2004/05', ['']),
 ('2004/05', ['3']),
 ('2005/06', ['']),
 ('2006/07', ['']),
 ('2007/08', ['']),
 ('2008/09', ['']),
 ('2009/10', ['']),
 ('2010/11', ['']),
 ('2011/12', ['']),
 ('2012/13', [''])]
```

In [120]:

```
list(zip(*(year_refs(dy) for dy in ads.index)))[0]
```

Out[120]:

```
('2002/03',
 '2002/03',
 '2002/03',
 '2002/03',
 '2003/04',
 '2003/04',
 '2004/05',
 '2004/05',
 '2005/06',
 '2006/07',
 '2007/08',
 '2008/09',
 '2009/10',
 '2010/11',
 '2011/12',
 '2012/13')
```

In [121]:

```
list(zip(*(year_refs(dy) for dy in ads.index)))[1]
```

Out[121]:

```
([''],
 ['2'],
 ['2', '4'],
 ['4'],
 [''],
 ['2'],
 [''],
 ['3'],
 [''],
 [''],
 [''],
 [''],
 [''],
 [''],
 [''],
 [''])
```

In [125]:

```
ads["Y"], ads["refs"] = list(zip(*(year_refs(dy) for dy in ads.index)))
```

In [126]:

```
ads
```

Out[126]:

	Total Expenses	Gross Earnings	Income Before Tax	Y	refs
Year					
2002/03	£134,108	£209,288	£75,180	2002/03	[]
2002/032	£65,700	£125,978	£60,278	2002/03	[2]
2002/032,4	£nan	£nan	£nan	2002/03	[2, 4]
2002/034	£nan	£nan	£nan	2002/03	[4]
2003/04	£137,227	£218,756	£81,529	2003/04	[]
2003/042	£69,950	£137,037	£67,087	2003/04	[2]
2004/05	£107,076	£199,352	£92,250	2004/05	[]
2004/053	£142,471	£nan	£nan	2004/05	[3]
2005/06	£131,615	£236,960	£105,346	2005/06	[]
2006/07	£135,812	£239,140	£103,328	2006/07	[]
2007/08	£141,027	£242,356	£101,330	2007/08	[]
2008/09	£148,388	£248,538	£100,175	2008/09	[]
2009/10	£142,627	£242,136	£99,518	2009/10	[]
2010/11	£145,445	£243,609	£98,173	2010/11	[]
2011/12	£147,391	£245,600	£98,209	2011/12	[]
2012/13	£149,318	£246,355	£97,073	2012/13	[]

In [128]:

```
ids = ads.reset_index()
ids
```

Out[128]:

	Year	Total Expenses	Gross Earnings	Income Before Tax	Y	refs
0	2002/03	£134,108	£209,288	£75,180	2002/03	[]
1	2002/032	£65,700	£125,978	£60,278	2002/03	[2]
2	2002/032,4	£nan	£nan	£nan	2002/03	[2, 4]
3	2002/034	£nan	£nan	£nan	2002/03	[4]
4	2003/04	£137,227	£218,756	£81,529	2003/04	[]
5	2003/042	£69,950	£137,037	£67,087	2003/04	[2]
6	2004/05	£107,076	£199,352	£92,250	2004/05	[]
7	2004/053	£142,471	£nan	£nan	2004/05	[3]
8	2005/06	£131,615	£236,960	£105,346	2005/06	[]
9	2006/07	£135,812	£239,140	£103,328	2006/07	[]
10	2007/08	£141,027	£242,356	£101,330	2007/08	[]
11	2008/09	£148,388	£248,538	£100,175	2008/09	[]
12	2009/10	£142,627	£242,136	£99,518	2009/10	[]
13	2010/11	£145,445	£243,609	£98,173	2010/11	[]
14	2011/12	£147,391	£245,600	£98,209	2011/12	[]
15	2012/13	£149,318	£246,355	£97,073	2012/13	[]

In [129]:

```
ids["Y"], ids["refs"] = list(zip(*(year_refs(dy) for dy in ids.Year)))
```

In [130]:

```
ids
```

Out[130]:

	Year	Total Expenses	Gross Earnings	Income Before Tax	Y	refs
0	2002/03	£134,108	£209,288	£75,180	2002/03	[]
1	2002/032	£65,700	£125,978	£60,278	2002/03	[2]
2	2002/032,4	£nan	£nan	£nan	2002/03	[2, 4]
3	2002/034	£nan	£nan	£nan	2002/03	[4]
4	2003/04	£137,227	£218,756	£81,529	2003/04	[]
5	2003/042	£69,950	£137,037	£67,087	2003/04	[2]
6	2004/05	£107,076	£199,352	£92,250	2004/05	[]
7	2004/053	£142,471	£nan	£nan	2004/05	[3]
8	2005/06	£131,615	£236,960	£105,346	2005/06	[]
9	2006/07	£135,812	£239,140	£103,328	2006/07	[]
10	2007/08	£141,027	£242,356	£101,330	2007/08	[]
11	2008/09	£148,388	£248,538	£100,175	2008/09	[]
12	2009/10	£142,627	£242,136	£99,518	2009/10	[]
13	2010/11	£145,445	£243,609	£98,173	2010/11	[]
14	2011/12	£147,391	£245,600	£98,209	2011/12	[]
15	2012/13	£149,318	£246,355	£97,073	2012/13	[]

In [ ]:

```

```