



Prifysgol Abertawe Swansea University

HEALTH CARE VISUALISATION User Manual

Team Members:

Robert Fletcher

Jonathan Phillips

Nicholas Rees

Contents

1	Introduction	2
2	Setup	3
2.1	Openshift	3
2.2	New Server	4
2.3	Customize	5
2.4	MySQL Import	5
3	Visualisation	7
3.1	Home Page	7
3.2	CCG Page	9
3.3	Main Visualisation	10
3.4	Other CCG charts	11
3.5	Practice Page	12

Chapter 1: Introduction

This is the user manual for the Healthcare Visualisation Project, which was undertaken with industrial partner, We Predict. This document will describe how to use the site and how to set it up on a new server.

The Project can be viewed here:

<http://wepredict.robrotheram.com/>

The site is separated in to 2 independent parts. The back-end, which is a Rest API which serves the data from the database, and the visualisation web application that is not dependent on the back-end at all, instead it uses the API produced by the back-end to generate visualisations from the data

Full Documentation for the project is available here:

<http://robrotheram.github.io/Documentation/>

Chapter 2: Setup

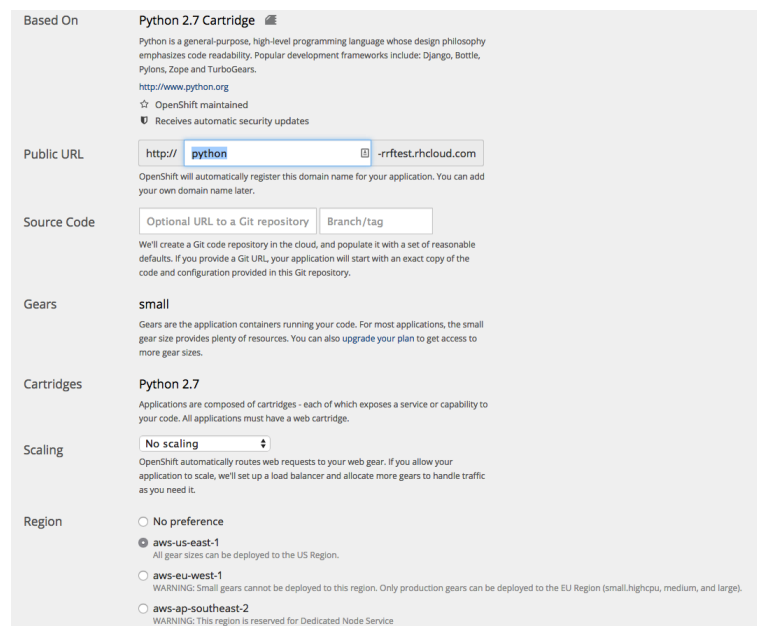
The project is designed to be very flexible with the setup, which means that there can be multiple combinations of technologies. For example the front end and the Back-end of the application can live on different servers. Or alternatively the back-end can be re-coded into a different language and so long as the URL paths remain the same only a minor change to the front-end will be needed.

2.1 Openshift

In the current state It is designed to work on Openshift. Openshift is a PaaS (Platform as a Service) service. The fastest way of getting the project running is to create an Openshift account, which is free for small applications.

<https://www.openshift.com/app/account/new>

Once an account has been created, the first thing to be done is to add a Python 2.7 application



The screenshot shows the 'Python 2.7 Cartridge' settings page in OpenShift. The page is divided into several sections:

- Based On:** Python 2.7 Cartridge. Description: Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Popular development frameworks include: Django, Bottle, Pylons, Zope and TurboGears. URL: <http://www.python.org>. Status: OpenShift maintained, Receives automatic security updates.
- Public URL:** <http://python-rrftest.rhcloud.com>. Note: OpenShift will automatically register this domain name for your application. You can add your own domain name later.
- Source Code:** Optional URL to a Git repository (empty) and Branch/tag (empty). Note: We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults. If you provide a Git URL, your application will start with an exact copy of the code and configuration provided in this Git repository.
- Gears:** small. Note: Gears are the application containers running your code. For most applications, the small gear size provides plenty of resources. You can also upgrade your plan to get access to more gear sizes.
- Cartridges:** Python 2.7. Note: Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.
- Scaling:** No scaling. Note: OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.
- Region:** ☒ aws-us-east-1 (All gear sizes can be deployed to the US Region). ☐ aws-eu-west-1 (WARNING: Small gears cannot be deployed to this region. Only production gears can be deployed to the EU Region (small,highcpu, medium, and large)). ☐ aws-ap-southeast-2 (WARNING: This region is reserved for Dedicated Node Service).

Figure 2.1: Openshift application settings

Following this, alter the settings for the git repository to:

<https://github.com/robtheram/wePredictAPI.git> and branch to master.

This will initially set it up with the git repository we have provided.

Once the application has been set up, the next step is to add a MySQL Database to the application. Once this is created the application and database will be properly set up. Then the server must be re-initialized to make sure that all dependencies are installed.

The next step is to set up the server on the Openshift platform:

1. Clone the REPO using the command *git clone URL of REPO - recursive*
2. Change the URL path to your server URL path in the file `wePredictAPI/static/front/app/js/services/wepredictapi.js`
3. Run the commands *git add.;git commit -m"init";git push*

This will set up the Server on Openshift. See the MySQL section for importing the Database. To get the setting for the MySQL database you are required to ssh into the server and enter the following:

```
echo $OPENSIFT_MYSQL_DB_HOST;  
echo $OPENSIFT_MYSQL_DB_PORT;  
echo $OPENSIFT_MYSQL_DB_USERNAME;  
echo $OPENSIFT_MYSQL_DB_PASSWORD;
```

2.2 New Server

The Project can be easily be configured to run on a custom server if using external hardware is not desired.

The server needs the following requisites:

- Python
- MySQL
- Apache with modProxy and Http Proxy also enabled, (nginx can also be used but you will need to write your own configuration route for this application)

First log into your server and navigate to where you want the application to live. Then clone the project using *git clone URL of REPO -recursive*

Next, it is important to change the URL path to your server, this is altered by changing the url path in the file `wePredictAPI/static/front/app/js/services/wepredictapi.js` Then change `wePredictAPI/settings.py` to your own MySQL server settings

For you convenience an Apache conf script has also been written. Just copy the `wePredict.conf` from `wePredictAPI/docs` to the apache sites-enabled directory and change the lines with `"/home/admin/python/wePredictAPI/"` for the path to the folder, and then restart apache

2.3 Customize

Both in Section 2.1 and 2.2 the entire project is running on the server. If you want it is possible to delete the front folder from `wepredictAPI/static/` and clone the visualisation web application repository and use it in another project or on a different server. All that is needed is the change to the URL in

`app/js/services/wepredictapi.js`

Since the server and frontend are decoupled it means that if for some reason you do not want to run a python server that so long that the output for each URL endpoint remains the same and that the URL structure also remains the same the front end visualisations will continue to work. Further documentation for the struture of the URLs and the json sturcture can be found here :

<http://wepredict.robrotheram.com/docs>

2.4 MySQL Import

This application would be nothing without the data. All that is needed for this project is available in SQL Scripts. These will create the database if it does not exist, creating and then imports all the data for that table.

Recommended Program: MySQL Workbench

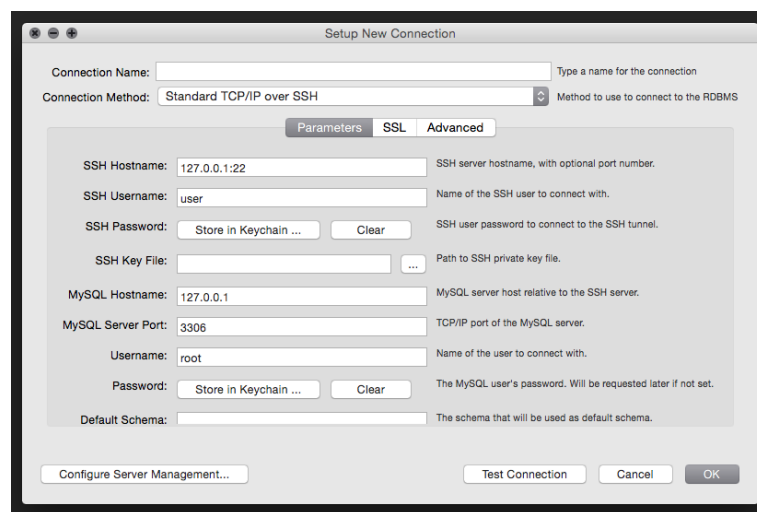


Figure 2.2: MySQL Workbench server settings

The first step with MySQL workbench is to log into the server. The most common way of doing this is to use the connection method of **over SSH pipe** if the MySQL server is running on a remote server.

Enter in all the settings to connect to the server, if you've used the suggested Openshift platform then set the path to **ssh key** then store it on your computer

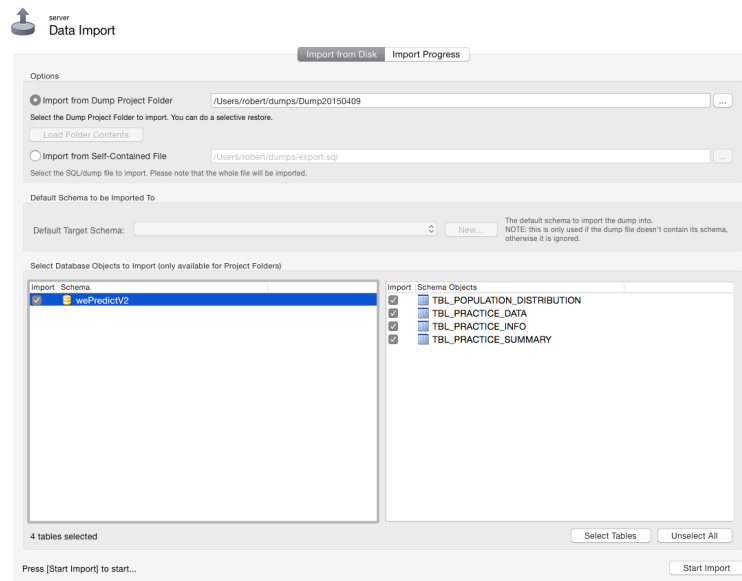


Figure 2.3: MySQL Workbench import SQL

Once connected to the server you can use the data import tool to import the database. In the tool navigate to the folder where the SQL files are and select all the tables. click import on the database tables and the data will be created. This process can take 5 - 20 minuets to complete depending on the server and connection.

Chapter 3: Visualisation

3.1 Home Page

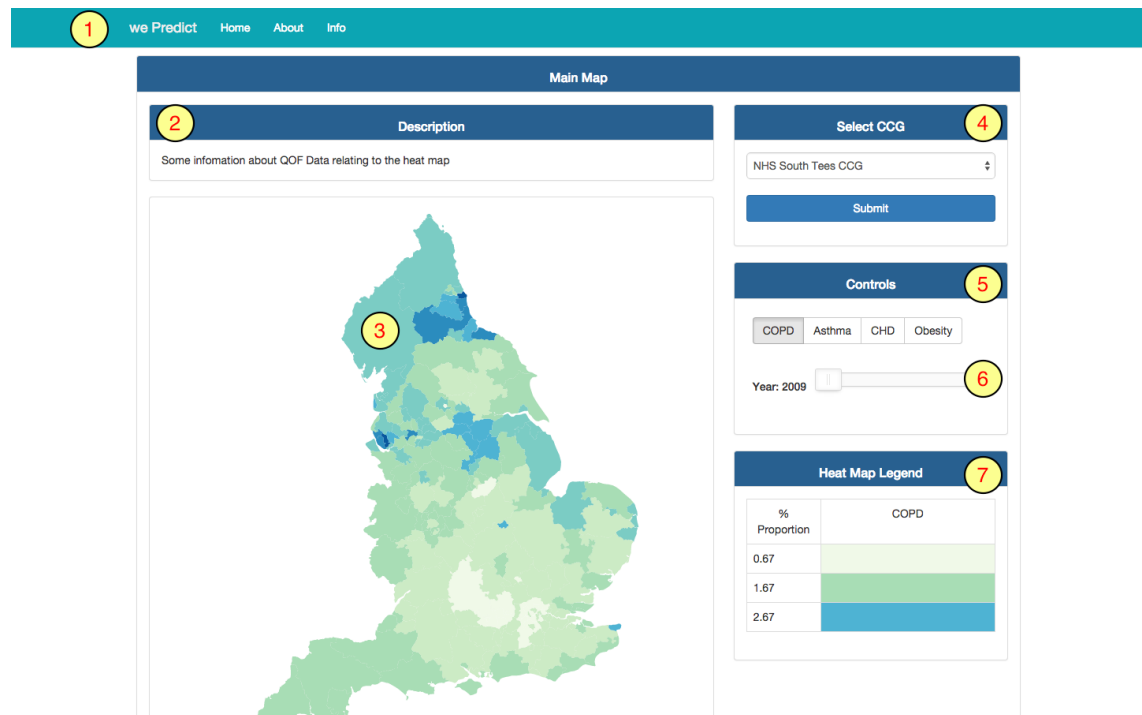
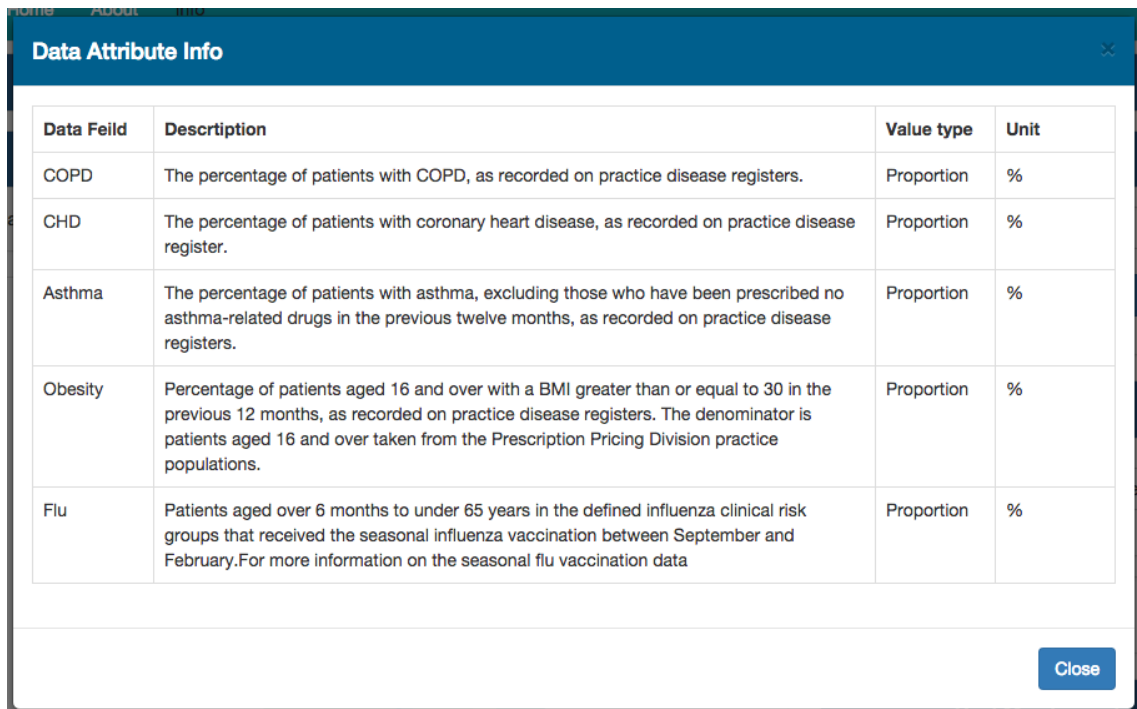


Figure 3.1:

1. Main menu bar
2. Description of the heat map updates with the CCG name when you hover over the heat map
3. Heat map updates with new data using the controls (5,6). Double clicking on an area on the map will take you to that CCG
4. Select box containing all CCGs in England. Select 1 and you will be taken to that view.
5. Control box, select the data field you want to display on the map
6. Slider, change the year of the data on the heat map.
7. Legend for the heat map, updates for each data field.



Data Field	Description	Value type	Unit
COPD	The percentage of patients with COPD, as recorded on practice disease registers.	Proportion	%
CHD	The percentage of patients with coronary heart disease, as recorded on practice disease register.	Proportion	%
Asthma	The percentage of patients with asthma, excluding those who have been prescribed no asthma-related drugs in the previous twelve months, as recorded on practice disease registers.	Proportion	%
Obesity	Percentage of patients aged 16 and over with a BMI greater than or equal to 30 in the previous 12 months, as recorded on practice disease registers. The denominator is patients aged 16 and over taken from the Prescription Pricing Division practice populations.	Proportion	%
Flu	Patients aged over 6 months to under 65 years in the defined influenza clinical risk groups that received the seasonal influenza vaccination between September and February. For more information on the seasonal flu vaccination data	Proportion	%

Figure 3.2:

If the user selects the info button near the top of the view, a module panel appears above the main content of the page. The panel contains all the information about the main data attributes, the value type and the unit for the data attribute.

3.2 CCG Page

The CCG page will be the main view that the user will use to analyse and compare the data. It centers on one main visualisation that will help users compare the CCG practices to see where a problem may occur. The page also contains other visualisations such as overview graphs at the bottom of the page. The page also contains a select option to drill down into a single practice.

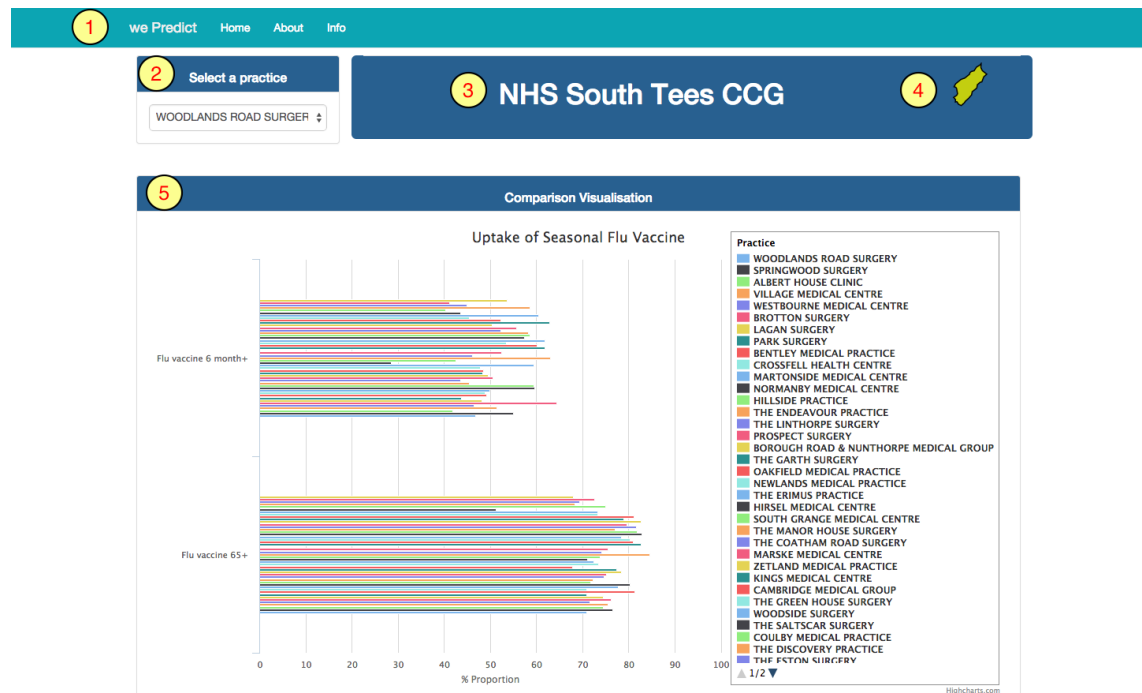


Figure 3.3:

1. Menu Menu
2. Select practice
3. Name of the practice
4. Visual outline of the CCG
5. Main visualisation

3.3 Main Visualisation

The main part of the CCG page is the manipulative chart section. In this visualisation you can set the chart type and data type from the drop down box below.

Other features of this visualisation are that a user can click on the hide all button which will remove all the data series from the graph. The a user can now click on an individual series from the legend so that you can compare several practices with each other.

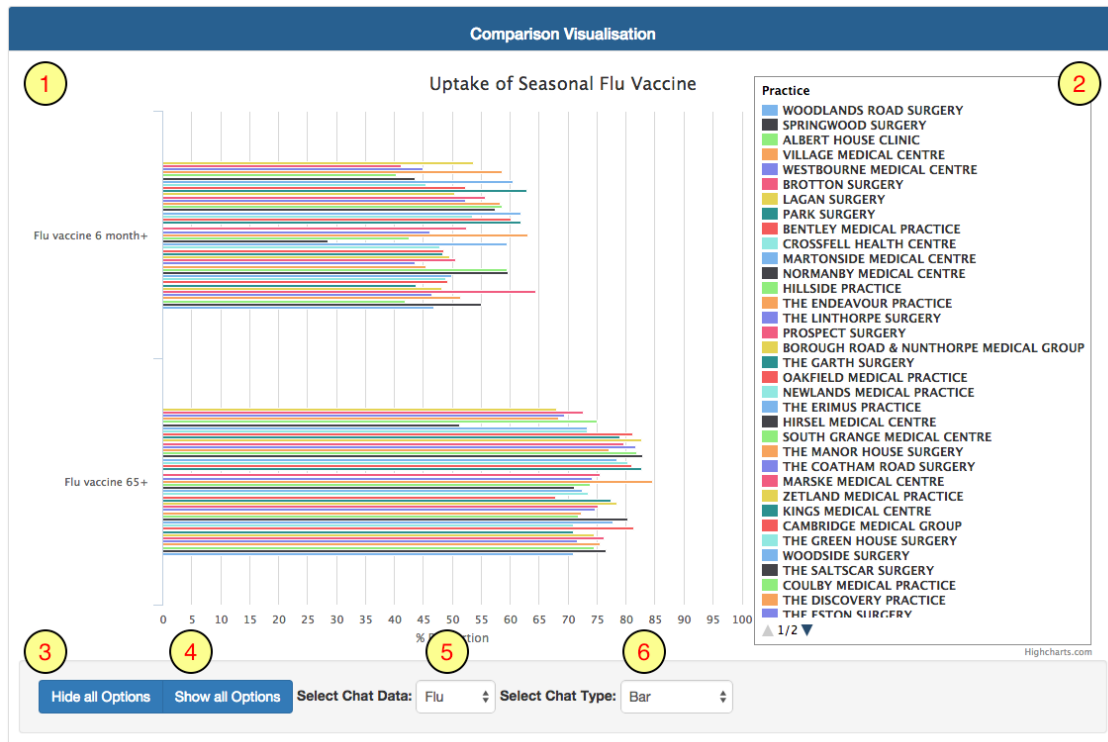


Figure 3.4:

1. Graph view of the visualisation
2. Legend of the visualisation, Clicking on each item in the Legend it will show or hide that field in the visualisation
3. button to hide all fields from the visualisation
4. button to Show all fields from the visualisation
5. Select which data attribute to visualise
6. Select what type of chart of the visualise

3.4 Other CCG charts

The CCG view contains a number of overview charts which rest at the bottom of the view. They contain information about the average data for all of the CCGs practices. The container that holds these panels can horizontally scroll revealing different data attributes.

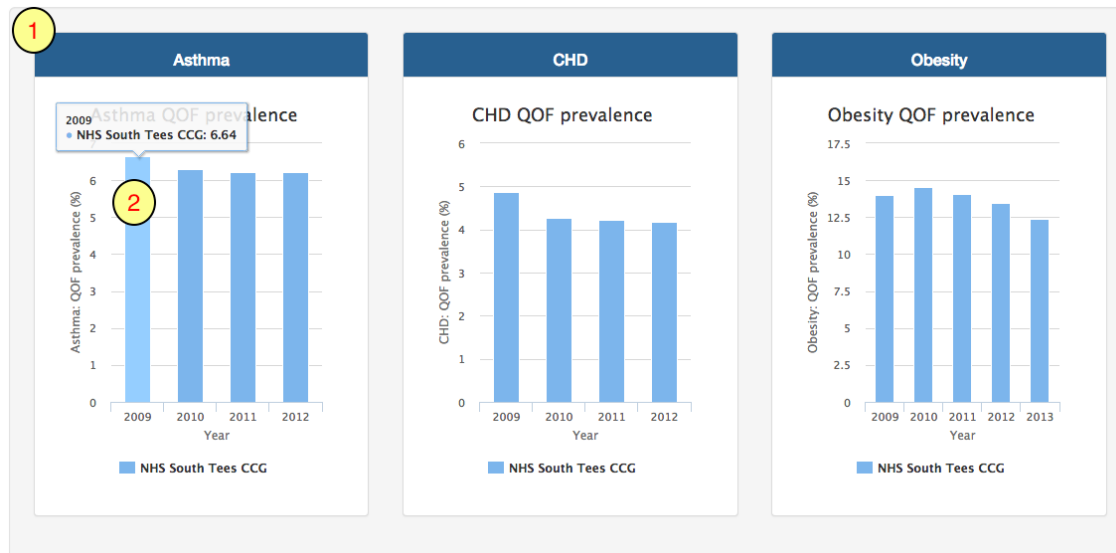


Figure 3.5:

1. Chart panel
2. Chart tool tip

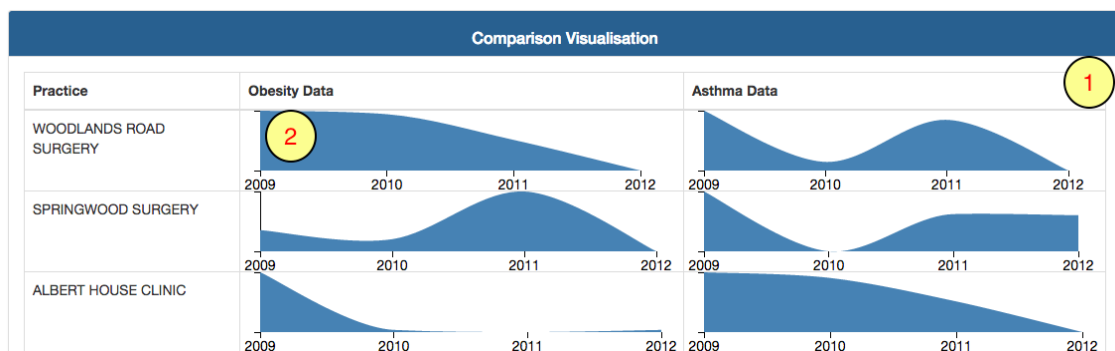


Figure 3.6:

The other graph on the CCG page is a small comparison visualisation. it leverages on a scrolling table to show each practice and the data. It allows for a quick overview glance to spot patterns or notice some anomaly in the data. The user may use the practice page to look at that information further.

1. Scrolling table of data
2. Graph for each series

3.5 Practice Page

The final section of this web application is the practice view. Here it allows a user to analyse a single practice. It Shows the CCG average for comparison. This page also allows you to change the practice to a different one in the CCG and it also allows the user to change the visualisation type since some data will be easier to understand using a different graph.

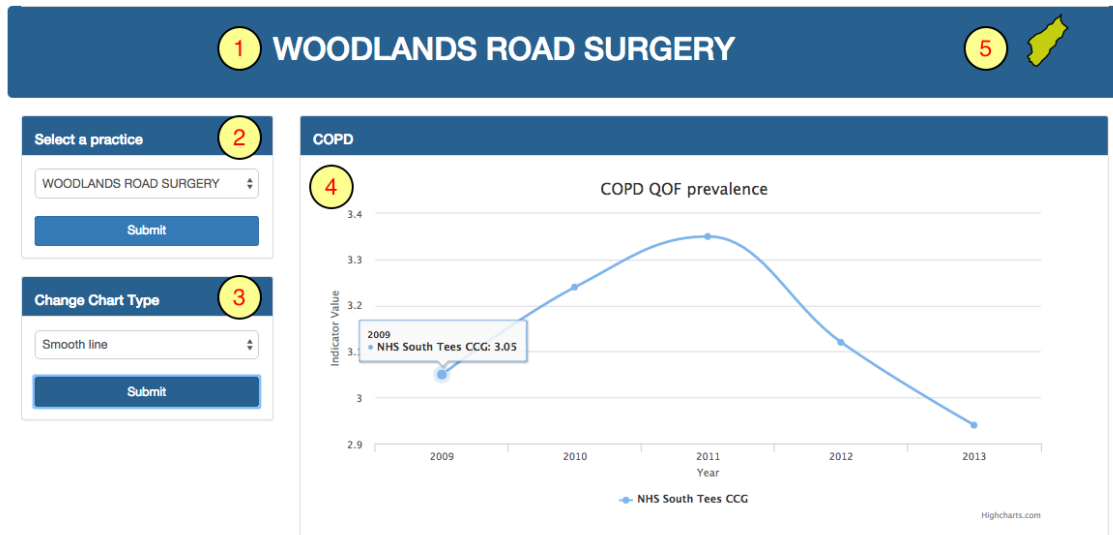


Figure 3.7:

1. Practice title
2. Select a different practice to view data from
3. Change all the charts to a different type
4. One chart shows the practice data against the CCG average.
5. Icon of the CCG