Crime Project - Methodology

# The Beginning

## Choosing The Data

After looking through some samples of both crime and stop and search data we quickly settled on using stop and search data. The data had numerous properties we thought would lead to useful analysis, and the quantity of records per period was lower so we could take data over a greater range of time and locations. We chose to analyse data for the last five years from the six constabularies that cover the five largest urban areas in the UK: The Metropolitan and City of London forces cover London; Greater Manchester for Manchester; the West Midlands covers Birmingham; Merseyside forces cover Liverpool; then for Leeds we looked at West Yorkshire. Unfortunately, the home office only gave us data going back to December 2014, undeterred we carried on thinking if there wasn’t enough data we would come back and take some more from different constabularies.

## Landing The Data

The Data from the Home Office was stored in folders marked by year and month, within each were separate csv files for each constabulary. Using SSIS we looped through the directory system and landed all the raw data into a table in an SQL database named crime.

## Profiling / EDA

Using SSIS’ profiling tools we looked at the raw data and gathered some key information on the columns their types, whether they contained discrete data, the length of the data involved. Two key findings were:

* There was a mixture of terms used for missing records
* Time wasn’t in a format appropriate for SQL
* There was no information about which constabulary carried out each stop

From this information we designed a database table system in 3rd normalised form, with appropriate column data types. In order to combat the points above we first decided we’d deal with missing values and time in SQL. Then for the last issue, we adjusted the SSIS package to introduce a ‘derived column’ that took the constabulary name from the file name and put it into all the records of that file.

## Munging

We decided to fill in all the fields which had missing values with the term ‘not stated’ for consistency, apart from the Latitude and Longitude columns which we kept as nulls in order to preserve the column type and avoid confusion with default values (SEE “Update-blanks-with-'not stated'.sql”). Now we set about the process of creating dimensions from the main table (SEE “Creating Dimensions” directory), where we dealt with the time issue mentioned earlier. We created surrogate keys to link between most dimensions, however for ethnicity there already existed a business key hence we made use of this instead.

# Analysis

After seeing some of the Tableau stories online I knew that this is what I wanted my finished project to be. The interaction between graphs, data and the user combined with the overall style made me think it would be perfect for the brief and a good skill to spend time learning. The aim was to create a tool for constabularies to use to help improve their effectivity.

## Location

It quickly became apparent that the latitude and longitude data provided by the home office was going to be incredibly problematic, if any analysis was going to be made by location. After speaking with colleagues it turned out that we had overlooked a crucial column in the disregarded crime data, LSOA codes. Codes that universally refer to small areas of the UK, consequently I set about making some link between the latitude and longitude data we possessed and the LSOA codes used in population data. After trying various avenues, the approach I settled on was the following:

1. Connecting each crime record to a postcode
   1. Import a table known as “dbo.dimLocation” which included outer postcodes (e.g. SW1), a central latitude, a central longitude, and the name of the town
   2. Create a function that accepts a latitude and longitude as parameters and searches through “dbo.dimLocation” to find which postcode is closest as the crow flies. (SEE “nearestPostcode-func.sql”)
   3. The function above isn’t terribly efficient so the next step I took was to bucket the latitude and longitude from each stop record, to three decimal places and get a distinct column of these.
   4. Then by applying the above function to said column I created a table that could act as an interim for each crime record supplying it with information above
2. Connecting LSOA codes to a postcode
   1. Import data on LSOA codes that had their full postcodes
   2. Use SQL to remove the second half of the postcodes
3. Import data on population by LSOA codes from the ONS
   1. Add the postcode column from step 2 onto this data

Now a link was established between each crime record, postcode, town name, and population data.

# Improvements

## Location

Key:

Crime

Postcode’s Latitude, Longitude

Diagram 1:

My function isn’t perfect, for example if you look at diagram 1 it’s easy to see how my function would incorrectly allocate the smaller postcode the crime. This could be rectified by either the police inputting the LSOA as they make stops, or I could pay for batch reverse geocoding of the data.

## Profiling

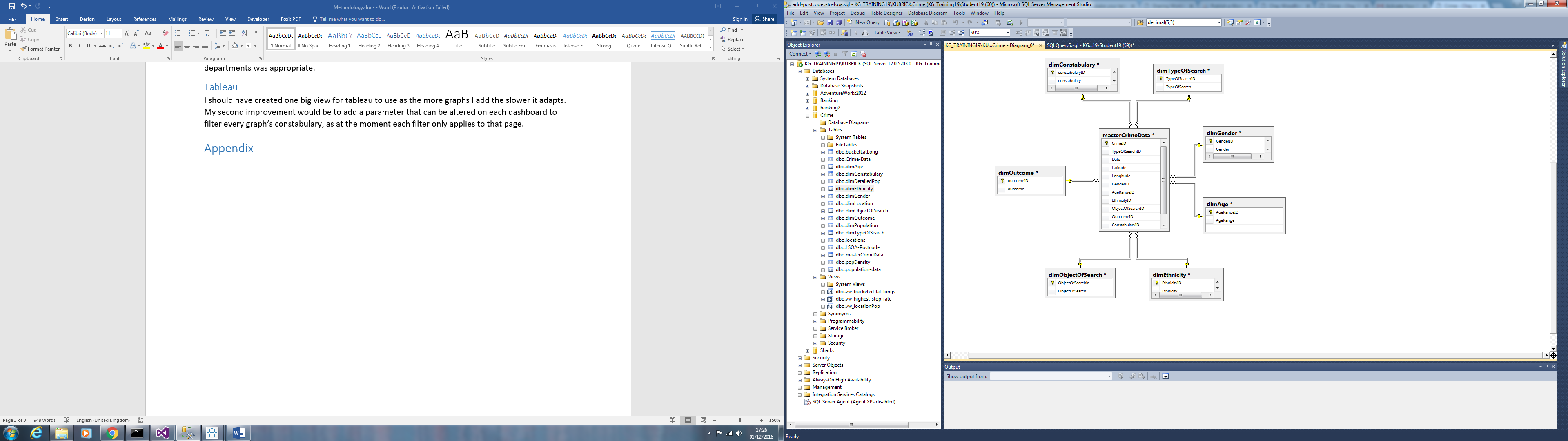
We should have profiled each constabulary’s data individually as the data taken varied greatly. In particular London only had location data for four months and hence no comparison between departments was appropriate.

## Tableau

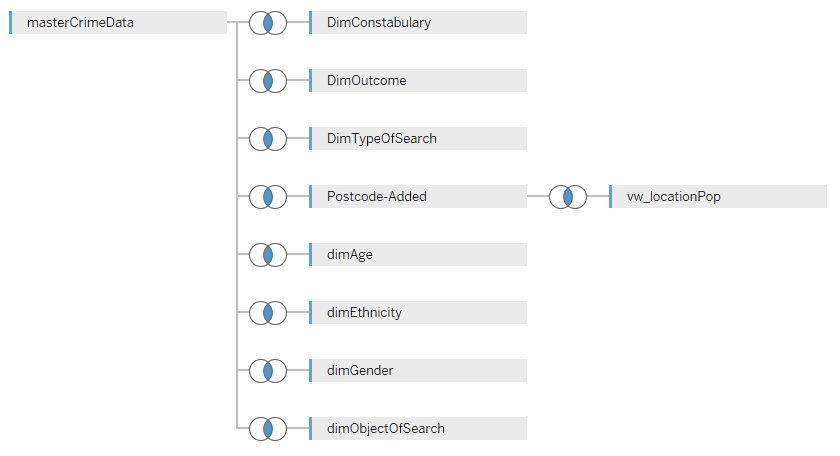
I should have created one big view for tableau to use as the more graphs I add the slower it adapts. My second improvement would be to add a parameter that can be altered on each dashboard to filter every graph’s constabulary, as at the moment each filter only applies to that page.

# Appendix

## Basic ERD



## Tableau Linking



## Postcode Added SQL

