# Where There is Life, There is Love

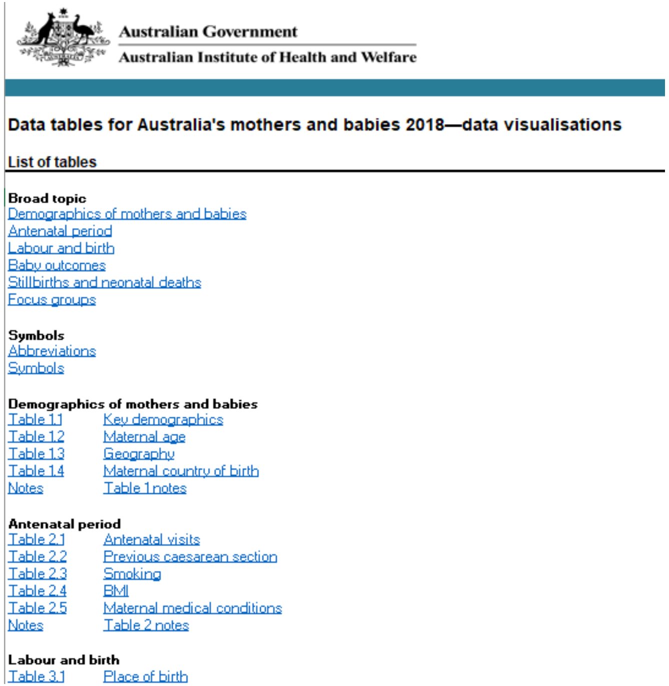
What am I going to do and what am I going to use for data?!

# Process

- Choose Data
- Decide on what questions I wanted to answer with the data
- Decide on technology stack to be used
- Start HTML design
- Reduce amount of data to come up with a new set of ‘final data’ to be analysed
- Start cleaning and investigating ‘final data’
- Start plotting/mapping
- Deployment
- More deployment
- More and more deployment
- Attempted deploying with Heroku, AWS, Azure, GoDaddy (paid hosting plan) then back to Heroku

# Data Sourcing & Decision Making

- Wanted data with multiple variables to enable interesting results
- Data source was in the form of excel workbooks
- Decided to break the data down into a more manageable size
- 'Final data' set, derived for further work
- 'Final data' was derived by going through the initial questions I had written down to answer when I started my project



Australian Government
Australian Institute of Health and Welfare

**Data tables for Australia's mothers and babies 2018—data visualisations**

List of tables

**Broad topic**
Demographics of mothers and babies
Antenatal period
Labour and birth
Baby outcomes
Stillbirths and neonatal deaths
Focus groups

**Symbols**
Abbreviations
Symbols

**Demographics of mothers and babies**

| | |
|---|---|
| Table 1.1 | Key demographics |
| Table 1.2 | Maternal age |
| Table 1.3 | Geography |
| Table 1.4 | Maternal country of birth |
| Notes | Table 1 notes |

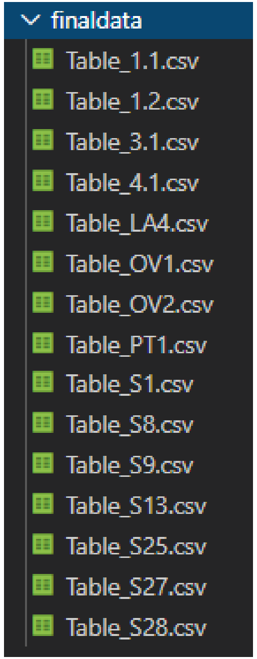
**Antenatal period**

| | |
|---|---|
| Table 2.1 | Antenatal visits |
| Table 2.2 | Previous caesarean section |
| Table 2.3 | Smoking |
| Table 2.4 | BMI |
| Table 2.5 | Maternal medical conditions |
| Notes | Table 2 notes |

**Labour and birth**

| | |
|---|---|
| Table 3.1 | Place of birth |

* 1 of 2 csv files downloaded



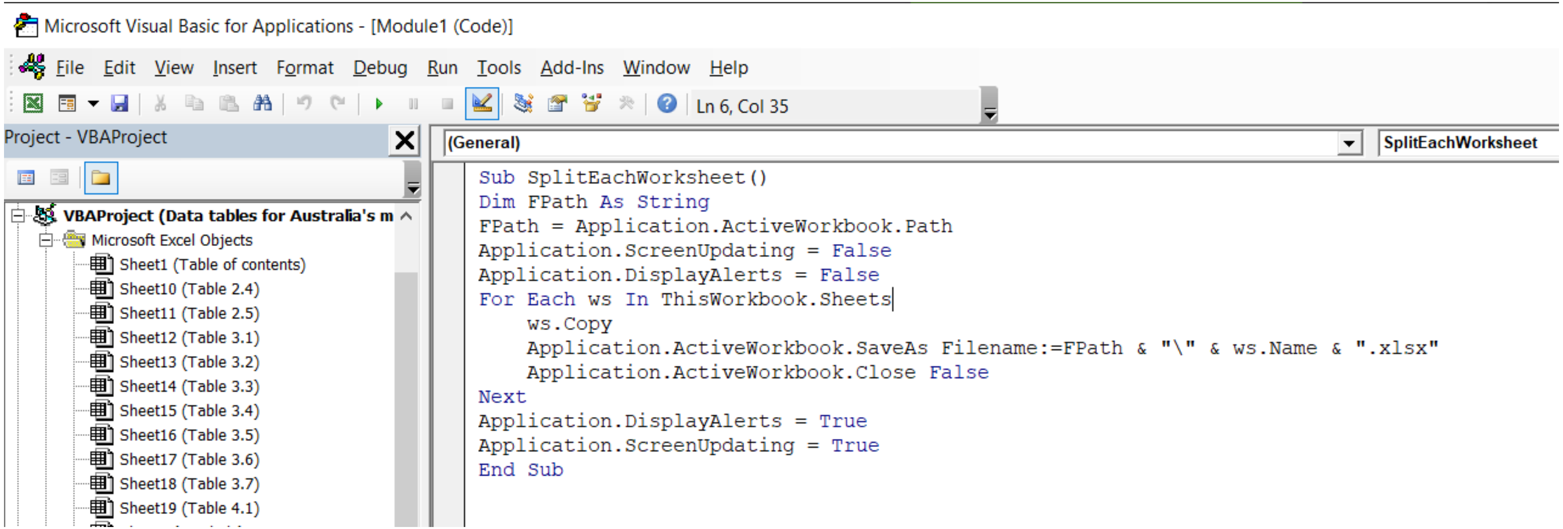
finaldata
Table_1.1.csv
Table_1.2.csv
Table_3.1.csv
Table_4.1.csv
Table_LA4.csv
Table_OV1.csv
Table_OV2.csv
Table_PT1.csv
Table_S1.csv
Table_S8.csv
Table_S9.csv
Table_S13.csv
Table_S25.csv
Table_S27.csv
Table_S28.csv

# Infrastructure

- VBA
- Heroku
- AWS
- PostgeSQL
- Flask
- Machine Learning – Linear Regression
- HTML/CSS
- Javascript
  - Leaflet.js
  - Owl Carousel
  - D3.js
  - AmCharts.js

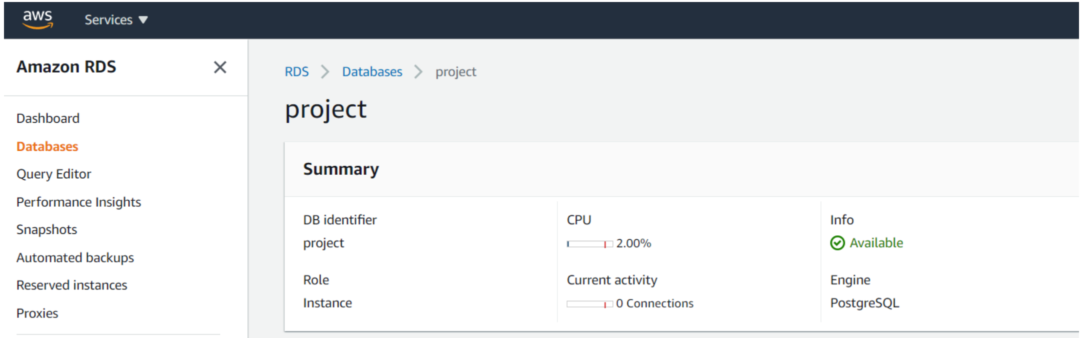
# VBA

VBA was really important for me within this project, it was utilised in order to split each of the worksheets into their own excel file, which was a huge time saving in order to analyse and then create a final set of data to be utilised.



```
Sub SplitEachWorksheet()
Dim FPath As String
FPath = Application.ActiveWorkbook.Path
Application.ScreenUpdating = False
Application.DisplayAlerts = False
For Each ws In ThisWorkbook.Sheets
    ws.Copy
    Application.ActiveWorkbook.SaveAs Filename:=FPath & "\" & ws.Name & ".xlsx"
    Application.ActiveWorkbook.Close False
Next
Application.DisplayAlerts = True
Application.ScreenUpdating = True
End Sub
```

# AWS & PostgreSQL



project

Summary

DB identifier: project

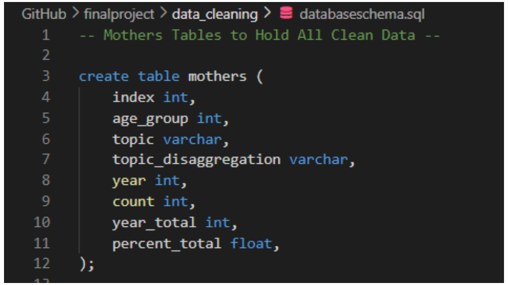
CPU: 2.00%

Info: Available

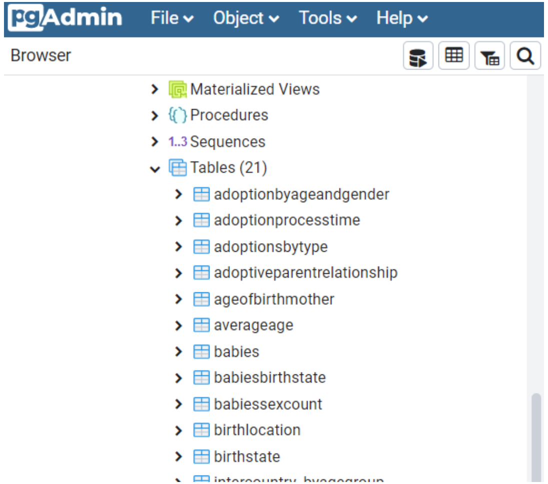
Role: Instance

Current activity: 0 Connections

Engine: PostgreSQL



```sql
-- Mothers Tables to Hold All Clean Data --

create table mothers (
    index int,
    age_group int,
    topic varchar,
    topic_disaggregation varchar,
    year int,
    count int,
    year_total int,
    percent_total float,
);
```



An AWS RDS instance was created in order to store my data, this was connected to my PostgreSQL Database in order to view and edit information with PGAdmin

# Flask

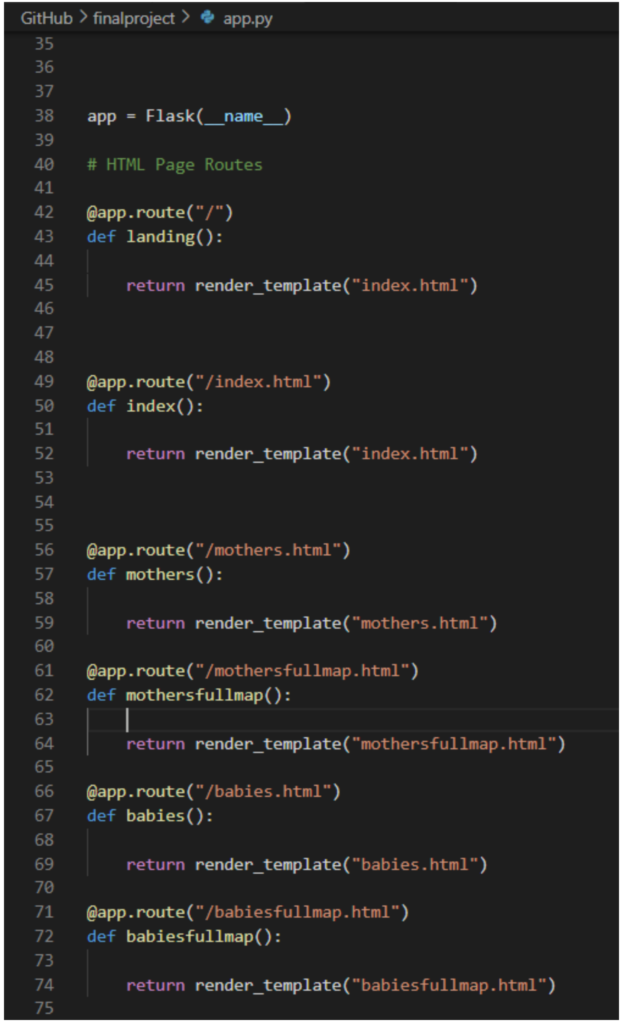
Flask is the one of the most important parts of my project.

## Flask Process – Initial – Working Locally

- Create the app which is able to render my HTML pages and link between each of them
- Be able to call my data from DB through my flask app to create my own API (and have the data come through in the correct format)
- Be able to access the data accurately to be able to plot

## Flask Process – Current – Deployed on Heroku

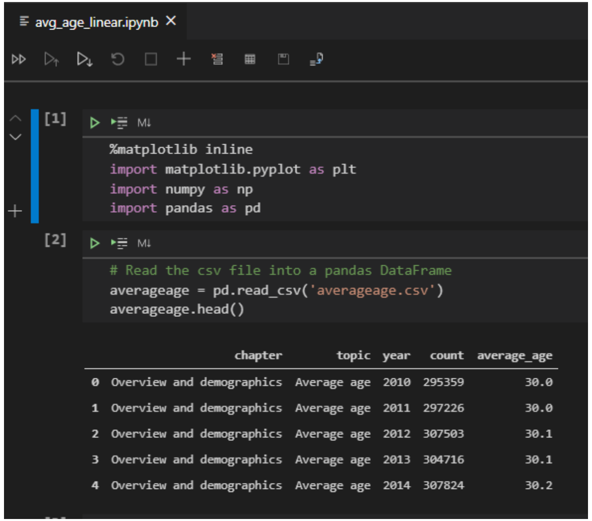
- App renders each of my HTML pages and links between each of them



```python



app = Flask(__name__)

# HTML Page Routes

@app.route("/")
def landing():

    return render_template("index.html")



@app.route("/index.html")
def index():

    return render_template("index.html")



@app.route("/mothers.html")
def mothers():

    return render_template("mothers.html")

@app.route("/mothersfullmap.html")
def mothersfullmap():

    return render_template("mothersfullmap.html")

@app.route("/babies.html")
def babies():

    return render_template("babies.html")

@app.route("/babiesfullmap.html")
def babiesfullmap():

    return render_template("babiesfullmap.html")

```

# Machine Learning – Linear Regression

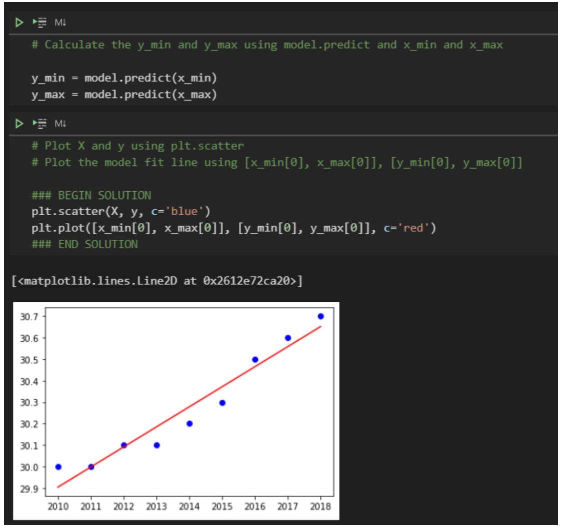
Linear Regression was used on one set of data, that was derived from a one excel file, taken from the 'final data' set. Simple machine learning model, but perfect for what I required, which was to be able to enter a year and to find out what the future average age of mothers would be in that year (or any year specified)



```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
# Read the csv file into a pandas DataFrame
averageage = pd.read_csv('averageage.csv')
averageage.head()
```

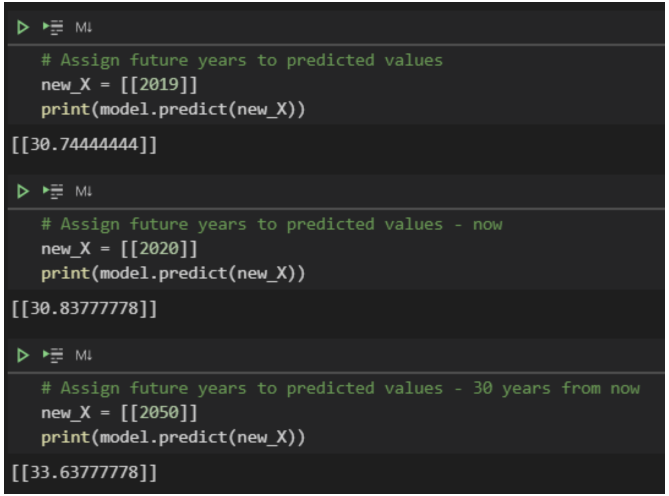
```
                      chapter        topic  year   count  average_age
0  Overview and demographics  Average age  2010  295359         30.0
1  Overview and demographics  Average age  2011  297226         30.0
2  Overview and demographics  Average age  2012  307503         30.1
3  Overview and demographics  Average age  2013  304716         30.1
4  Overview and demographics  Average age  2014  307824         30.2
```



```
# Calculate the y_min and y_max using model.predict and x_min and x_max

y_min = model.predict(x_min)
y_max = model.predict(x_max)
```

```
# Plot X and y using plt.scatter
# Plot the model fit line using [x_min[0], x_max[0]], [y_min[0], y_max[0]]

### BEGIN SOLUTION
plt.scatter(X, y, c='blue')
plt.plot([x_min[0], x_max[0]], [y_min[0], y_max[0]], c='red')
### END SOLUTION
```

```
[<matplotlib.lines.Line2D at 0x2612e72ca20>]
```



```
# Assign future years to predicted values
new_X = [[2019]]
print(model.predict(new_X))
```

```
[[30.74444444]]
```

```
# Assign future years to predicted values - now
new_X = [[2020]]
print(model.predict(new_X))
```

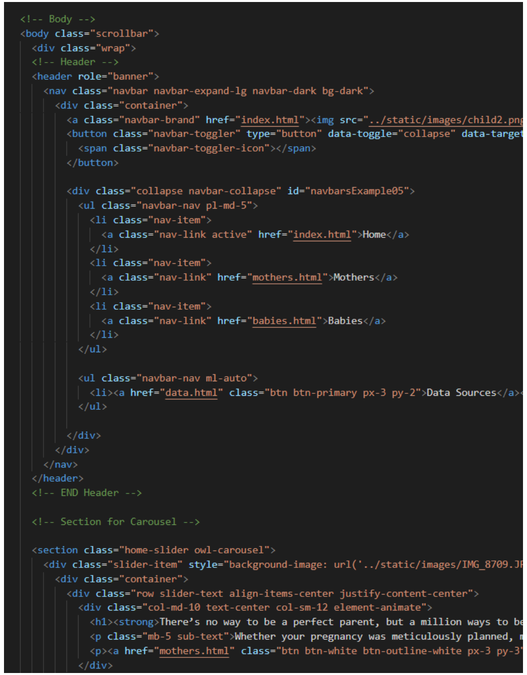
```
[[30.83777778]]
```

```
# Assign future years to predicted values - 30 years from now
new_X = [[2050]]
print(model.predict(new_X))
```

```
[[33.63777778]]
```

# HTML/CSS/JavaScript

Aside from Flask, the HTML/CSS and JavaScript component(s) of my project are the most important to bring everything together.



```html
<!-- Body -->
<body class="scrollbar">
  <div class="wrap">
  <!-- Header -->
  <header role="banner">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container">
        <a class="navbar-brand" href="index.html"><img src="../static/images/child2.png
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target
          <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarsExample05">
          <ul class="navbar-nav pl-md-5">
            <li class="nav-item">
              <a class="nav-link active" href="index.html">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="mothers.html">Mothers</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="babies.html">Babies</a>
            </li>
          </ul>

          <ul class="navbar-nav ml-auto">
            <li><a href="data.html" class="btn btn-primary px-3 py-2">Data Sources</a>
          </ul>

        </div>
      </div>
    </nav>
  </header>
  <!-- END Header -->

  <!-- Section for Carousel -->

  <section class="home-slider owl-carousel">
    <div class="slider-item" style="background-image: url('../static/images/IMG_8709.JP
      <div class="container">
        <div class="row slider-text align-items-center justify-content-center">
          <div class="col-md-10 text-center col-sm-12 element-animate">
            <h1><strong>There’s no way to be a perfect parent, but a million ways to be
            <p class="mb-5 sub-text">Whether your pregnancy was meticulously planned, m
            <p><a href="mothers.html" class="btn btn-white btn-outline-white px-3 py-3"
          </div>
```

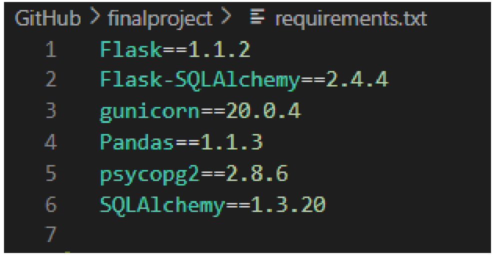
The HTML framework is based on a couple of key libraries, Owl Carousel, and Bootstrap.

The charts utilise D3.js, amCharts.js and the maps are from Leaflet.js

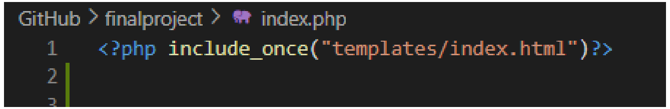
# Heroku Deployment

**There were a few key parts to my Heroku Deployment**

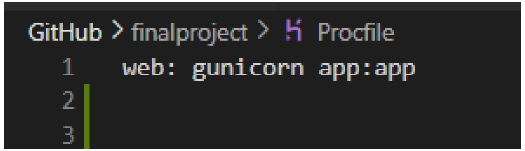
- **requirements.txt** file, to specify all of the packages required to be installed in Heroku when initialising the app
- **index.php file**, in order to trick Heroku our app is a PHP Application.
  - Hypertext Preprocessor – PHP is a scripting language
- **Procfile**, to specify the commands that are executed by the app on startup
- **composer.json** file, specifys dependancies required, however even if there is none, a blank {} is required, to be recognised as a PHP application



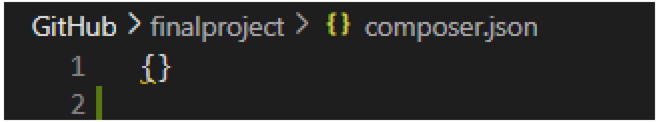
```
Flask==1.1.2
Flask-SQLAlchemy==2.4.4
gunicorn==20.0.4
Pandas==1.1.3
psycopg2==2.8.6
SQLAlchemy==1.3.20
```



```
<?php include_once("templates/index.html")?>
```



```
web: gunicorn app:app
```



```
{}
```

[Site Hosted on Heroku]

# Thanks!

## Any questions?

You can find me at:

https://github.com/KirstieMcCown

https://www.linkedin.com/in/kirstie-mccown/