

INTERNSHIP REPORT

Data Science in the Real World

PI Prescott (2020).

THE TASK

The student will work on a research project defined between the student and an external organisation.

The aims of the placement will be defined in terms of *progressive risks in effecting a solution*:

- The **first aim** should have a *low risk of failure*;
- the **second aim** will be *more challenging but capable of solution given initiative and energy on the part of the student*;
- and the **third aim** can have a 'blue skies' element, *a real research challenge* and consequently a high risk of failure but success will demonstrate exceptional competence and initiative.

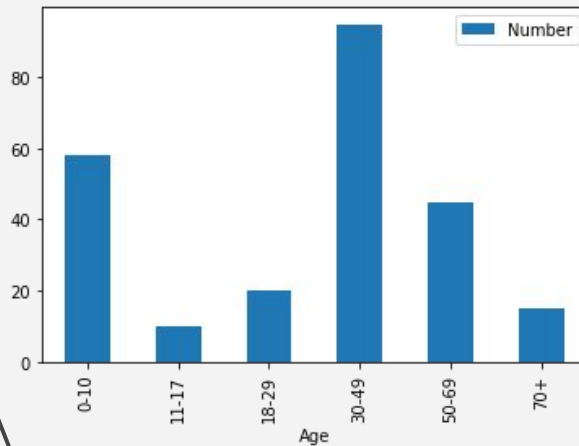
https://vital.liv.ac.uk/webapps/UoL-uol-module-overview-BBLEARN/module/view.jsp?course_id= 861125_1&mode=view



THE PARTNER



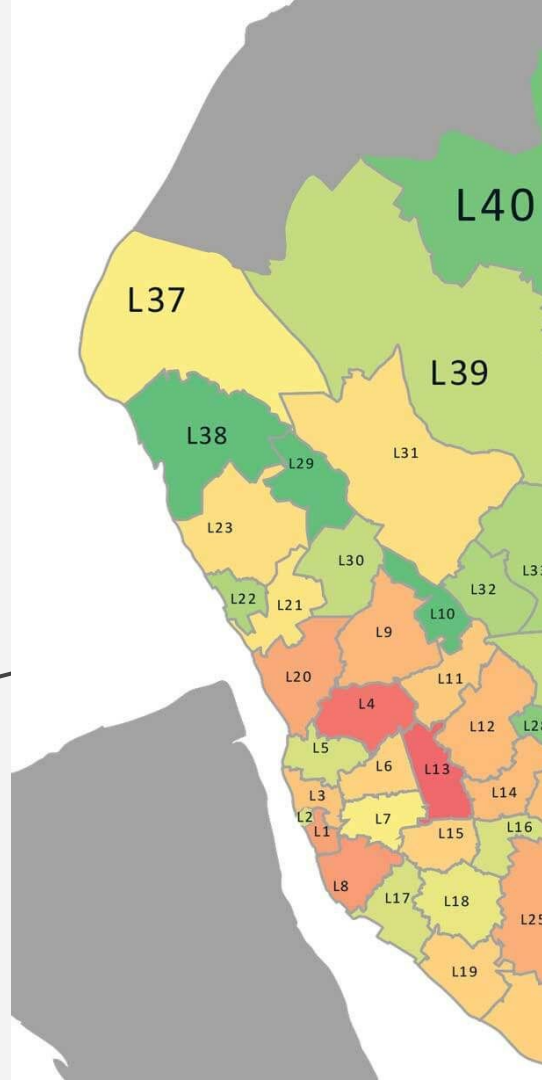
Mossley Hill Church
A Community Of Blessing



Led by the Rev Alan Kennedy.
Attended by Prof Alex Singleton.
A family church in south Liverpool.
An Anglican parish church,
with notable Victorian architecture.

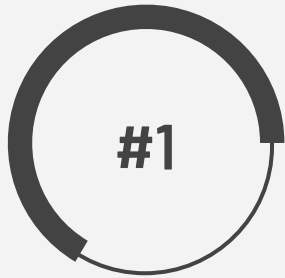
RESEARCH PROJECT

To transform the
church members'
postcode data into
useful insight.



The Three Aims:

Neighbourhood
Geodemographics



#1

The analysis of people
by where they live.

Catchment Area
Analysis



#2

The distance people will
come for a service.

Developing a
Data Dashboard



#3

Individualized, instantaneous,
interactive informational insight.



NEIGHBOURHOOD GEODEMOGRAPHICS

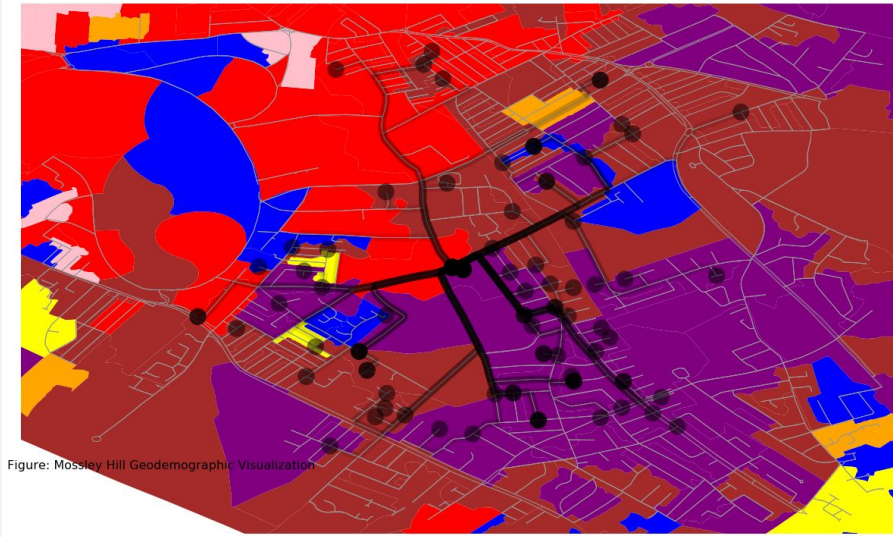
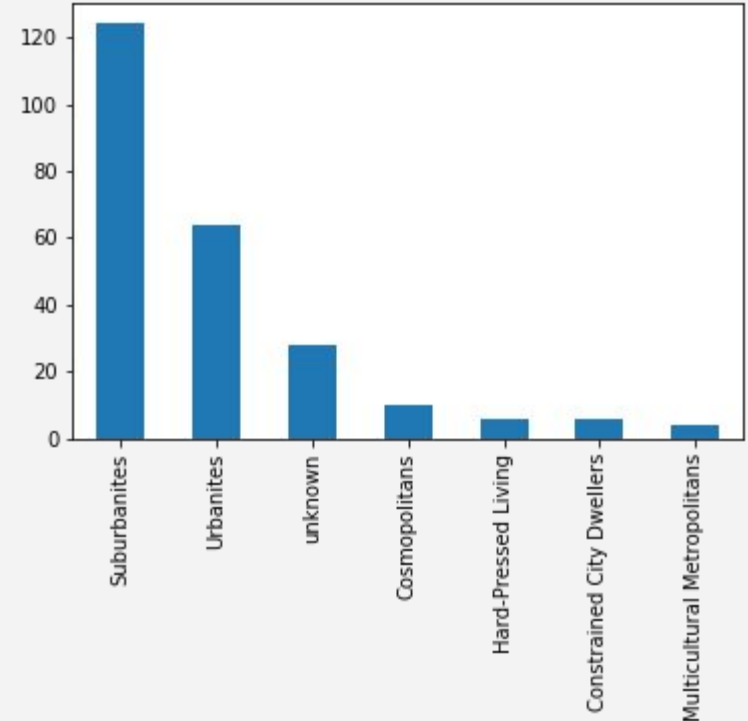
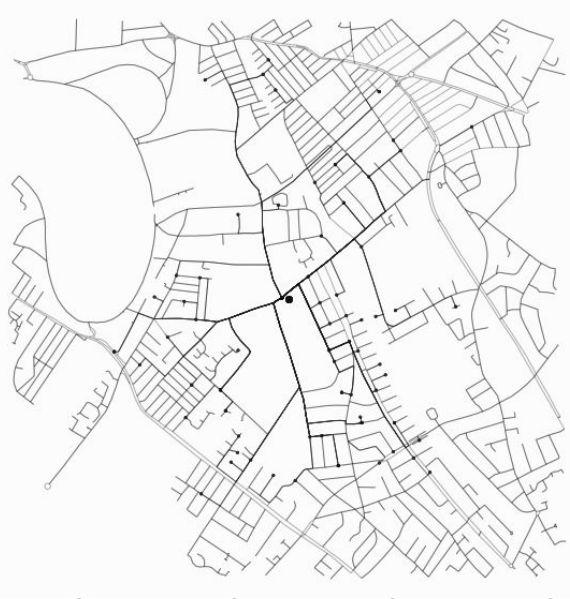
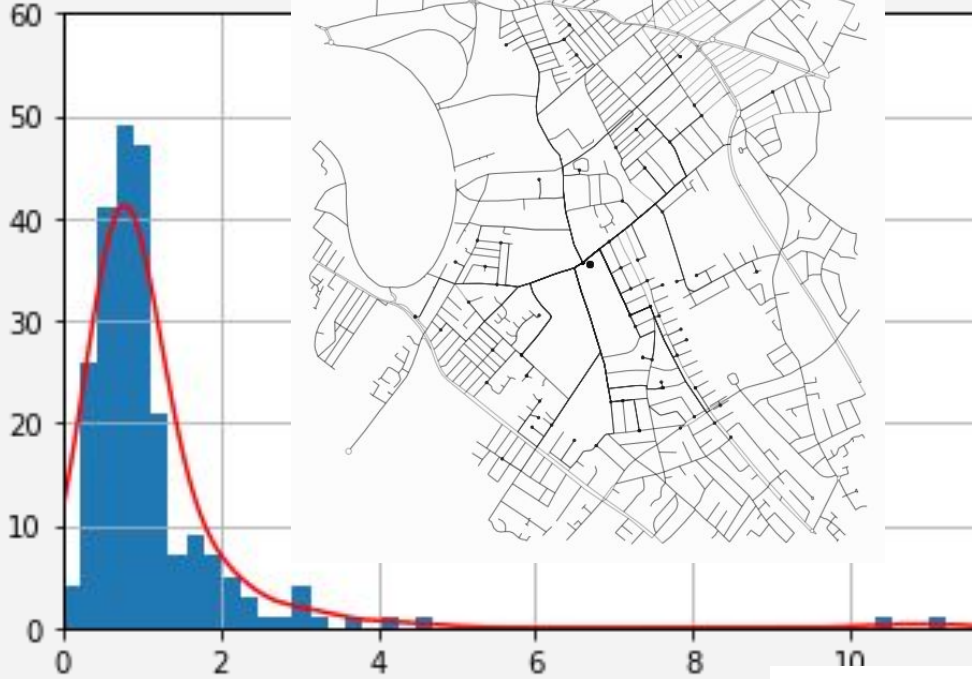


Figure: Mossley Hill Geodemographic Visualization



Analysis using the free and open-source
2011 Output Area Classification

CATCHMENT AREA ANALYSIS



Analysis used Geoff Boeing's OSMnx Python package.

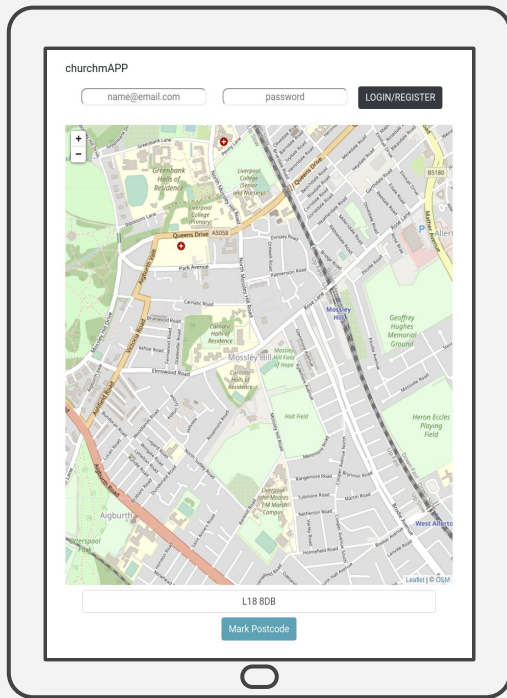
```
1 | import osmnx as ox  
2 | ox.plot_graph(ox.graph_from_place('
```

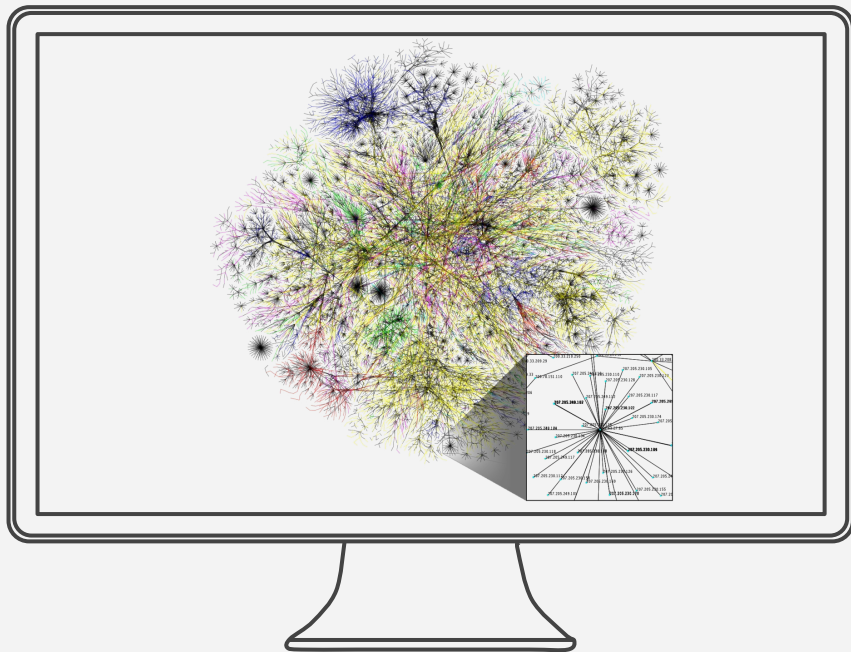

<https://churchmAPP.netlify.com>

DEVELOPING A DATA DASHBOARD

__churchmAPP__

a Full-Stack Modern Web App



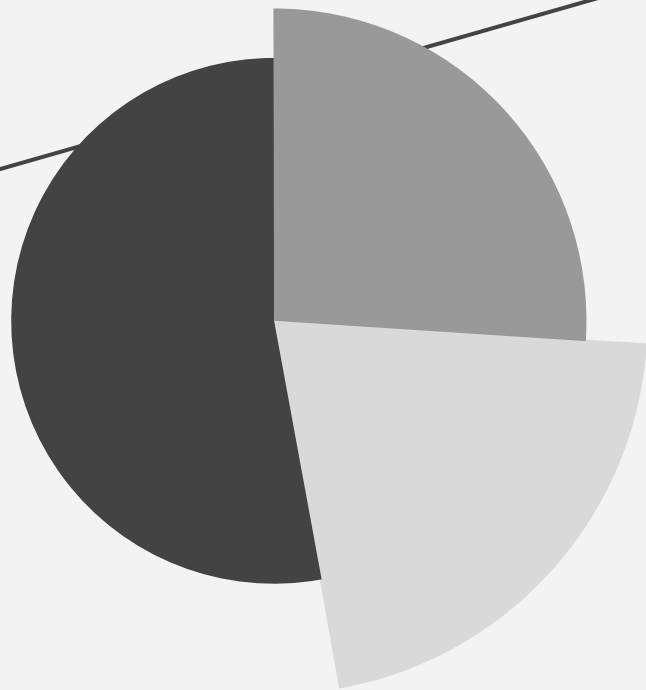


What is the internet?

- 750,000 miles of cable
- Internet (computers) vs. WWW (documents)
- Hypertext: HTTP(S) and HTML
- Status Codes: 404, 500, (418!)
- HTML, CSS, JavaScript
- Front-end, Back-end, APIs

“The internet is a series of tubes.” (Senator Ted Stevens)

Modern Web Frameworks



Flask (Python Web Server)



SQLAlchemy (SQL ORM)

Flask-Restful (ReST API)



Bootstrap (CSS)

Vue (Javascript DOM)



Leaflet (Javascript Maps)



Data Security

The Elements:

- Authentication
- Authorization
- Encryption
- Crypto-Hashing
- Risk Management

The Implementation:

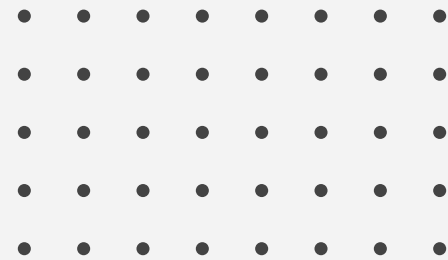
- SHA256 Hash
- JSON Web Tokens

<https://churchmapp.netlify.com/auth>

```
29 ## user registration/authentication
30 @app.route('/auth', methods=['POST'])
31 def authenticate():
32     try:
33         email = request.json['email']
34         password = request.json['password']
35
36         user = session.query(User)
37             .filter(User.email==email)
38             .first()
39         if user: # If user already exists...
40             # ... authenticate password
41             if not pbkdf2_sha256
42                 .verify(
43                     password,
44                     user.password_hash
45                 ):
46                 return jsonify(
47                     {'message': 'Wrong password for that email'},
48                     401
49                 )
50             else:
51                 message = 'Authentication successful'
52
53         else: # else create new password hash and user
54             hash = pbkdf2_sha256.hash(password)
55             user = User(email=email, password_hash=hash)
56             session.add(user)
57             session.commit()
58             message = 'Registration successful'
59
60         # everyone still here gets a token
61         payload = {'email': email }
62         token = jwt.generate_jwt(
63             payload, JWT_security_key,
64             'PS256', timedelta(minutes=15))
65         return jsonify({
66             'message': message,
67             'JWT': token
68         }), 200
69
70 except Exception as e:
71     return jsonify({'error': str(e)[:100]}), 500
72
73 ## JWT validation
74 @app.route('/validate', methods=['POST'])
75 def validate():
76     try:
77         token = request.headers.get('JWT')
78         header, claims = jwt.verify_jwt(token, JWT_security_key, ['PS256'])
79         email = claims['email']
80         payload = {'email': email}
81         new_token = jwt.generate_jwt(
82             payload, JWT_security_key,
83             'PS256', timedelta(minutes=15))
84         message = 'Token validation successful'
85
86         return jsonify({'message': message,
87             'JWT': new_token}), 200
88
89 except Exception as e:
90     return jsonify({'message': 'Validation failed'}), 401
```

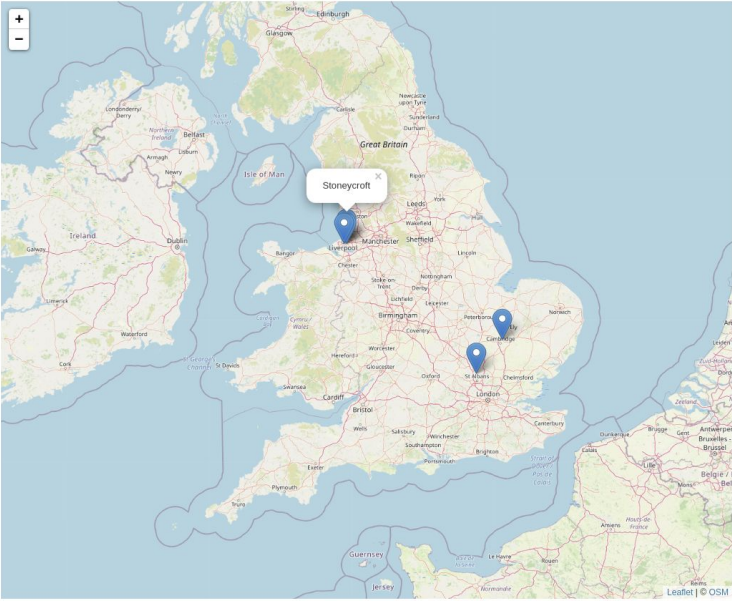


Modern DevOps Best Practices



- Source Control (git, GitHub, GitLab)
- Configuration as Code (YAML/TOML)
- Automated Testing (Jupyter Notebooks -- !)
- Immediate Release (Netlify, PythonAnywhere)
- Continual Deployment (webhooks)

<https://github.com/peterprescott/churchmapp>

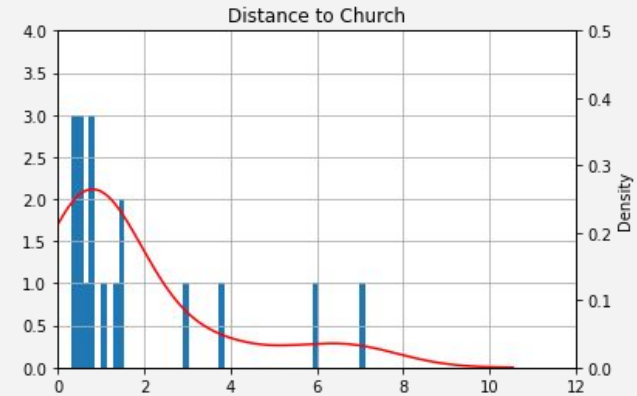


CB4 0DS

Mark Postcode

Name	Postcode	[X]
Mossley Hill Church	L18 8DB	X
the University (?)	L69 3BX	X
Stoneycroft	L13 7EQ	X
I used to work here	AL5 4BX	X
Undergrad	CB4 0DS	X

From Mossley Hill to the
uttermost ends of the earth...





THANKS!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

