SQL_Script

209041841

07/03/2021

GY7708 Geospatial Databases and Information Retrival

SQL Queries

The University of Leicester

Coursework 1

RMarkdown and SQL

Load library

Create a connection

Connection Established.

Question 1

In Question 1, We have to write the necessary SQL code to execute the queries assigned in the table, The queries as follows:

Query 2: Retrieve the date of birth of Wes Anderson.

```
--Select the person name and date of birth from wikidata_htji_person

SELECT
  person_name, date_of_birth
FROM
  wikidata_htji_person
WHERE
    person_name = 'Wes Anderson'
    AND date_of_birth is NOT NULL;
```

Table 1: .

person_name	date_of_birth
Wes Anderson	1969-05-01

Query 4: Retrieve the total boxoffice takings per year of release.

```
-- Extract the year by using the date_part function and select the relase year and
-- total the boxoffice amount using sum function.

SELECT
    date_part('year',release_date) AS year,
    sum(boxoffice) AS total_boxoffice
FROM
    wikidata_htji_movie
GROUP BY
  year
ORDER BY
  year
LIMIT 10;
```

Table 2: .

year	total_boxoffice
1922	2500000
1930	10637000
1931	5751000
1932	8000000
1938	951801
1940	5000000
1950	16581000
1952	6500000
1953	17138000
1957	27200000

Query 9: Retrieve the number of genres associated with the movie “Pacific Rim”.

```
-- Identify the genre that are affilated to the movie "Pacific Rim".

SELECT
    whm.movie_title,
    m_genere.genre_name
FROM
    (
    SELECT
        whmg.*,
        whg.genre_name
    FROM
        wikidata_htji_movie_genre AS whmg
    INNER JOIN wikidata_htji_genre AS whg ON
        whmg.genre_id = whg.genre_id ) AS m_genere
LEFT OUTER JOIN wikidata_htji_movie AS whm ON
    m_genere.movie_id = whm.movie_id
WHERE
    whm.movie_title = 'Pacific Rim'
LIMIT 10;
```

Table 3: .

movie_title	genre_name
Pacific Rim	action film
Pacific Rim	adventure film
Pacific Rim	superhero film
Pacific Rim	alien invasion
Pacific Rim	speculative fiction film
Pacific Rim	action film
Pacific Rim	adventure film
Pacific Rim	superhero film
Pacific Rim	alien invasion
Pacific Rim	speculative fiction film

Query 11: Retrieve the average boxoffice takings by movies with Jason Isaacs.

```
-- Calculate the average box office movies collection acted by Jason Isaacs.

SELECT
    m_actor.person_name,
    round(avg(whm.boxoffice)) AS average_boxoffice
FROM
    (
    SELECT
        whma.*,
        whp.person_name
    FROM
        wikidata_htji_movie_actor AS whma
    INNER JOIN wikidata_htji_person AS whp ON
        whma.person_id = whp.person_id ) AS m_actor
LEFT OUTER JOIN wikidata_htji_movie AS whm ON
    m_actor.movie_id = whm.movie_id
WHERE
    m_actor.person_name = 'Jason Isaacs'
GROUP BY
    m_actor.person_name;
```

Table 4: .

person_name	average_boxoffice
Jason Isaacs	353017693

Query 15: retrieve the list of directors who worked on a movie with a title including the word "Hotel", and for each director the average runtime of all their movies.

```
-- Calculate average run timing of movie directed by movie which contains the word "Hotel"

SELECT
```

```
    m_director.person_name,
    avg(whm.duration) AS average_duration
FROM
    (
    SELECT
        whmd.*,
        whp.person_name,
        whp.primary_profession
    FROM
        wikidata_htji_movie_director AS whmd
    LEFT OUTER JOIN wikidata_htji_person AS whp ON
        whmd.person_id = whp.person_id
    WHERE
        whp.primary_profession = 'director'
    ORDER  BY
        person_name ) AS m_director

INNER JOIN wikidata_htji_movie AS whm ON
    m_director.movie_id = whm.movie_id
WHERE
    movie_title LIKE '%Hotel%'
    OR movie_title LIKE '%hotel%'
GROUP BY
    m_director.person_name;
```

Table 5: .

person_name	average_duration
Genndy Tartakovsky	97
Wes Anderson	99

Question 2

Task - 1 : Analysing the redundancies in the CSV raw data file

Creating the new database for the information containing about the england musuem along with respective Local Authority District(LAD) details from the Wikidata_Museums_Eng_LAD.csv.

The dataset wiki_museums_eng_lad.csv contains the museums data in the England, this dataset includes the museums name, latitude, longitude, respective alter names of the museums along with the local area authority which has the information about the area code, area name, authority type, government office and its head of institution also the inception date, population and the area of the LAD. After detailed review of the raw data, there is a wealth of knowledge about the LAD and the museum, but there are redundancies.

The data is typically too cluttered to upload in a single database, and it often has a higher number of null values. Some of the data has several duplicate entries that must be extracted from the table in order to operate effectively. If we upload this data to a database server, it would not be feasible. To make the data more essential, we must create three separate tables that will allow all users to easily access the data.

The first table, wikidata_eng_lad_authority, will contain information specific to LAD data such as lad_code, lad_name, authority_type, government_office and government_head. The primary key for this table will be lad_code, which will aid users in relating to other tables. We're developing a new table called

wikidata_lad_pop_area that contains the lad_code, inception_date, population, and area in order to make LAD data more accessible. where lad_code operates as the table's primary key, allowing it to be connected to other tables. The third table, wikidata_museum_location_lad, will only contain information about museums in England. This table will include museum_name, longitude, latitude, alternate names, and lad_code. Where museum act as the primary key and lad_code can be used as the foreign key with reference to the table wikidata_eng_lad_authority. We can connect all three tables together by using a primary key and a foreign key.

We can resolve redundant entries in the raw data and eliminate duplication in the data by generating these three tables. Now that we've converted messy data into content that's easy to navigate, these tables will upload it to the database server. We'll use a SQL script to build three separate tables in the sections below.

Task - 2 : Creation of tables in SQL

```
create_table <- st_read(
  pgsql_conn,
  query = "

  -- Drop table if already exists_____________________________________

  DROP TABLE IF EXISTS public.wikidata_eng_lad_authority CASCADE;
  DROP TABLE IF EXISTS public.wikidata_lad_pop_area;
  DROP TABLE IF EXISTS public.wikidata_museum_location_lad;

  -- LAD Authority table_____________________________________________

  CREATE TABLE public.wikidata_eng_lad_authority
  (
  lad_code varchar NULL,
  lad_name varchar NULL,
  authority_type varchar NULL,
  government_office varchar NULL,
  government_head varchar NULL,
  CONSTRAINT wikidata_eng_lad_authority_pk PRIMARY KEY (lad_code)
  );

  -- LAD population and area table___________________________________

  CREATE TABLE public.wikidata_lad_pop_area
  (
  lad_code varchar NULL,
  Lad_inception_date timestamp NULL,
  lad_population int8 NULL,
  lad_area double precision NULL,
  CONSTRAINT wikidata_lad_pop_area_pk PRIMARY KEY (lad_code)
  );

  -- Museum location table___________________________________________

  CREATE TABLE public.wikidata_museum_location_lad
  (
  museum_name varchar NOT NULL,
  longitude double precision NOT NULL,
```

```
  latitude double precision NOT NULL,
  alternate_names varchar NULL,
  lad_code varchar NULL,
  CONSTRAINT wikidata_museum_location_lad_pk PRIMARY KEY (museum_name),
  CONSTRAINT wikidata_museum_location_lad_fk FOREIGN KEY (lad_code)
             REFERENCES wikidata_eng_lad_authority (lad_code)
  );
"
)
```

Now, the tables has been created. We will insert the values into the table in below code.

Task - 3 : Inserting values into the created tables

```
insert_rows <- st_read(
  pgsql_conn,
  query = "

--Table values of wikidata_eng_lad_authority_____________________________________

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E07000047','West Devon','non-metropolitan district',null,null);

INSERT INTO
  wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E06000054','Wiltshire','unitary authority of England',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E06000057','Northumberland','unitary authority of England',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E07000151','Daventry','non-metropolitan district',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E08000023','South Tyneside','metropolitan borough',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E07000178','Oxford','non-metropolitan district',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
```

```
government_head)
VALUES ('E08000025','Birmingham','metropolitan borough',
'leader of Birmingham City Council', null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E07000179','South Oxfordshire','non-metropolitan district',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E07000077','Uttlesford','non-metropolitan district',null,null);

INSERT INTO
    wikidata_eng_lad_authority (lad_code,lad_name,authority_type,government_office,
government_head)
VALUES ('E08000037','Gateshead','metropolitan borough',null,null);

-- Table values of wikidata_lad_pop_area_____________________________________________

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E07000047', null, null, null);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E06000054',null,null,null);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E06000057',null,null,null);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E07000151','1974-04-01T00:00:00Z',84484,662.6157);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E08000023','1974-04-01T00:00:00Z',150265,64.3943);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E07000178',null,null,null);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E08000025','1974-04-01T00:00:00Z',1141374,267.7913);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E07000179','1974-04-01T00:00:00Z',137400,678.5372);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E07000077',null,null,null);

INSERT INTO wikidata_lad_pop_area (lad_code,lad_inception_date,lad_population,lad_area)
VALUES ('E08000037','1974-04-01T00:00:00Z',202508,142.3593);

-- Table values of wikidata_museum_location_lad_____________________________________
```

```
INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Buckland Abbey',-4.13361,50.4811,'abbaye de Buckland;Abbazia di Buckland',
'E07000047');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Lacock Abbey',-2.11718,51.41475,'abbaye Sainte-Marie de Lacock;abbazia di Lacock',
'E06000054');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Alnwick Castle',-1.706078,55.415783,'château de Alnwick;Alnwick Castle',
'E06000057');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Althorp',-1.00207,52.2802,'Althorp;Althorp','E07000151');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Castra Arbeia',-1.43,55.004,'Arbeia','E08000023');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Ashmolean Museum',-1.26056,51.75556,'Ashmolean Museum;Ashmolean Museum',
'E07000178');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Aston Hall',-1.88436,52.5063,'Aston Hall;Aston Hall','E08000025');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Aston Martin Heritage Trust Museum',-1.13723,51.66,
'Aston Martin Heritage Trust Museum;Aston Martin Heritage Trust Museum',
'E07000179');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Audley End House',0.220555556,52.02083333,'Audley End House;Audley End House',
'E07000077');

INSERT INTO
    wikidata_museum_location_lad (museum_name,longitude,latitude,alternate_names,lad_code)
VALUES ('Baltic Centre for Contemporary Art',-1.597777778,54.96916667,
'Baltic Center for Contemporary Art','E08000037');
"
)
```

Now, values are inserted into the tables. we can view the tables in below:

```
--To view Museum details
```

```
SELECT *
FROM wikidata_museum_location_lad;
```

Table 6: .

museum_name	longitude	latitude	alternate_names
Buckland Abbey	-4.1336100	50.48110	abbaye de Buckland;Abbazia di Buckland
Lacock Abbey	-2.1171800	51.41475	abbaye Sainte-Marie de Lacock;abbazia di Lacock
Alnwick Castle	-1.7060780	55.41578	château de Alnwick;Alnwick Castle
Althorp	-1.0020700	52.28020	Althorp;Althorp
Castra Arbeia	-1.4300000	55.00400	Arbeia
Ashmolean Museum	-1.2605600	51.75556	Ashmolean Museum;Ashmolean Museum
Aston Hall	-1.8843600	52.50630	Aston Hall;Aston Hall
Aston Martin Heritage Trust Museum	-1.1372300	51.66000	Aston Martin Heritage Trust Museum;Aston Martin Herit
Audley End House	0.2205556	52.02083	Audley End House;Audley End House
Baltic Centre for Contemporary Art	-1.5977778	54.96917	Baltic Center for Contemporary Art

```
--To view LAD authority details

SELECT *
FROM wikidata_eng_lad_authority;
```

Table 7: .

lad_code	lad_name	authority_type	government_office	government_head
E07000047	West Devon	non-metropolitan district	NA	NA
E06000054	Wiltshire	unitary authority of England	NA	NA
E06000057	Northumberland	unitary authority of England	NA	NA
E07000151	Daventry	non-metropolitan district	NA	NA
E08000023	South Tyneside	metropolitan borough	NA	NA
E07000178	Oxford	non-metropolitan district	NA	NA
E08000025	Birmingham	metropolitan borough	leader of Birmingham City Council	NA
E07000179	South Oxfordshire	non-metropolitan district	NA	NA
E07000077	Uttlesford	non-metropolitan district	NA	NA
E08000037	Gateshead	metropolitan borough	NA	NA

```
--To view LAD population an d area

SELECT *
FROM wikidata_lad_pop_area;
```

Table 8: .

lad_code	lad_inception_date	lad_population	lad_area
E07000047	NA	NA	NA
E06000054	NA	NA	NA
E06000057	NA	NA	NA
E07000151	1974-04-01	84484	662.6157
E08000023	1974-04-01	150265	64.3943
E07000178	NA	NA	NA
E08000025	1974-04-01	1141374	267.7913

lad_code	lad_inception_date	lad_population	lad_area
E07000179	1974-04-01	137400	678.5372
E07000077	NA	NA	NA
E08000037	1974-04-01	202508	142.3593

Load library

Create a connection for shared database

Question 3: Spatial analysis of cinemas in greater london area.

Task - 1 : Proposing spatial analysis.

The London output area classification (LOAC) was the methodology used to create from 2011 OAC census data by Office of the National Statistics. The LOAC categorized the people living area into eight different supergroups such as

A. Intermediate Lifestyle, B. High Density and High Rise Flats, C. Settled Asians, D. Urban Elites, E. City Vibe, F. London Life-Cycle,
G. Multi-Ethnic Suburbs, H. Ageing City Fringe

These categories are more categorized into 15 subgroups further for better understanding for the London community. We will use greater_london_osm_polygon, greater_london_osm_point, greater_london_osm_line, and greater_london_osm_line from the Open Street Map (OSM) and greater_london_loac in this study to look at the number of cinemas in Greater London and the urban landscape in which they are situated. The point and polygon tables contain different details about buildings, addresses, amenities, highways, and so on, while the line table will cover the greater London road network. The London supergroups and subgroups will be covered by the LOAC table. The Since London is a cosmopolitan area, the majority of residents rely on public transit for their mode of transportation. People are often entertained on weekends in order to spend more time with their families, most people's first preference is perhaps the theatre or a restaurant. We will assess the ease of access to cinemas (i.e., bus stop and train station) as well as the gap between nearby restaurants and the road access to the cinemas in the Greater London area in this study. London is one of the most visited cities in the world by tourists from all over the world. Tourism is one of the government's most significant sources of income, we will extract the most famous cinema hall with tourists. London has one of the best railways and roadway networks in the world; we'll compare this network to the number of cinema halls in the country, and we'll measure the number of people in each category in the greater London area. To achieve the desired results, we will use the SQL queries mentioned below.

```
-- Calculating distance from railway station to polygons cinemas

SELECT
    id,
    name,
    ST_Transform( 'SRID=4326;POINT(-0.1477611 51.4528669)'::geometry,27700)
    <->
    ST_Transform(geom,27700) AS distance
FROM
    greater_london_osm_polygon
WHERE
    amenity = 'cinema'
ORDER BY
```

```
    distance
LIMIT 10;
```

Table 9: .

id	name	distance
8036	Streatham Odeon	2630.478
586693	Capital Studios	3268.319
83745	Fulham Road Picturehouse	4384.748
329755	ODEON Putney	4771.035
64017	East Dulwich Picturehouse Cinema	4965.997
50015	Peckham Plex	5863.579
424916	Rooftop Film Club	5870.005
49237	Curzon Mayfair	5945.013
96318	BFI IMAX	6224.419
124227	Empire Cinemas	6291.654

```
-- Calculate the distance from railway station for points

SELECT
    osm_id ,
    name,
    ST_Transform( 'SRID=4326;POINT(-0.1477611 51.4528669)'::geometry,27700)
    <->
    ST_Transform(geom,27700) AS distance
FROM
    greater_london_osm_point
WHERE
    other_tags = '"amenity"=>"cinema"'
ORDER BY
    distance
LIMIT 10;
```

Table 10: .

osm_id	name	distance
498694180	Cineworld	4040.061
1716046769	Imperial Cinema	5691.848
442518406	Coronet	5777.625
1833349017	Cineworld Shaftesbury Avenue	6545.109
25474641	Barbican Centre (Cinema)	8334.597
300614361	Odeon Cinema	9566.512
264759489	Empire Sutton	10343.327
1097177019	Watermans Arts Centre	11056.402
420752260	Empire	12256.645
4690153399	ArtHouse Crouch End	14473.077

```
-- Calculate the bustation available for polygons

SELECT
```

```
    ply1.name AS Cinema,
    ply2.name AS Bus_Station
FROM
    greater_london_osm_polygon AS ply1,
    greater_london_osm_polygon AS ply2
WHERE
    ply1.amenity = 'cinema'
    AND ply2.amenity = 'bus_station'
    AND ST_DWITHIN( ST_Transform(ply1.geom,27700),
    ST_Transform(ply2.geom,27700),2000)
LIMIT 10;
```

Table 11: .

cinema	bus_station
VUE Cenima	Hammersmith Bus Station
VUE Cenima	Hammersmith Bus Station
Cineworld Enfield	NA
Genesis Cinema	Aldgate Station
Genesis Cinema	Mile End Station
Odeon Shaftesbury Avenue	NA
NA	Hammersmith Bus Station
Curzon Mayfair	Victoria Coach Station (Arrivals)
Curzon Mayfair	Victoria Coach Station (Departures)
Curzon Mayfair	Victoria Bus Station

```
-- Calculate bustop available for point table cinemas

SELECT
    pty1.name AS Cinema,
    pty2.name AS Bus_Station
FROM
    greater_london_osm_point AS pty1,
    greater_london_osm_point AS pty2
WHERE
    pty1.other_tags = '"amenity"=>"cinema"'
    AND pty2.highway = 'bus_stop'
    AND ST_DWITHIN( ST_Transform(pty1.geom,27700),
    ST_Transform(pty2.geom,27700),2000)
ORDER BY
    pty2.highway ASC
LIMIT 10;
```

Table 12: .

cinema	bus_station
Coronet	Pocock Street
Barbican Centre (Cinema)	Fleet Street / City Thameslink
Barbican Centre (Cinema)	Liverpool Street Station
Watermans Arts Centre	Victoria Gate
Watermans Arts Centre	Lion Gate
Watermans Arts Centre	Brentford Gate

cinema	bus_station
Watermans Arts Centre	Queen Charlotte's Cottage
ArtHouse Crouch End	Sunnyside Road
ArtHouse Crouch End	Mulkern Road
ArtHouse Crouch End	Hazellville Road

```
-- Calculate nearest restaurant available for cinemas

SELECT
    poly1.name AS cinema,
    poly2.name AS restaurant
FROM
    greater_london_osm_polygon AS poly1,
    greater_london_osm_polygon AS poly2
WHERE
    poly1.amenity = 'cinema'
    AND poly2.amenity = 'restaurant'
    AND ST_DWITHIN( st_transform(poly1.geom,27700),
    st_transform(poly2.geom,27700),100)
ORDER BY
    poly2.amenity ASC
LIMIT 10;
```

Table 13: .

cinema	restaurant
Odeon Shaftesbury Avenue	Wildwood Kitchen
Cineworld	Nando's
Rooftop Film Club	Nim's Kitchen
Curzon Mayfair	Iran Restaurant
Empire Cinemas	Norte
Crouch End Picturehouse	Paccata
Crouch End Picturehouse	The Belash India
ArtHouse Crouch End	The Belash India
Crouch End Picturehouse	Bistro Aix
ArtHouse Crouch End	Bistro Aix

```
-- Calculate nearest restaurant available for point table cinemas

SELECT
    point1.name AS cinema,
    point2.name AS restaurant
FROM
    greater_london_osm_point AS point1,
    greater_london_osm_point AS point2
WHERE
    point1.other_tags = '"amenity"=>"cinema"'
    AND point2.other_tags = '"amenity"=>"restaurant"'
    AND ST_DWITHIN( ST_Transform(point1.geom,27700),
    ST_Transform(point2.geom,27700),100)
```

```
ORDER BY
    point2.other_tags ASC
LIMIT 10;
```

Table 14: .

cinema	restaurant
Empire	Pho Saigon
Empire	Chicos
Cineworld Shaftesbury Avenue	Thai Tho
Cineworld Shaftesbury Avenue	XU
Cineworld Shaftesbury Avenue	Orient
Cineworld Shaftesbury Avenue	Hot Pot
Cineworld Shaftesbury Avenue	Sophie's Steak House
Cineworld Shaftesbury Avenue	Pizza Toscana

```
-- Find the cinema hall attaracted by the tourists.

SELECT
    ply.name,
    ply.tourism
FROM
    greater_london_osm_polygon AS ply
WHERE
    amenity = 'cinema'
    AND tourism = 'attraction';
```

Table 15: .

name	tourism
BFI IMAX	attraction

```
-- Calculate the number supergroup areas in greater london

--A  Intermediate Lifestyle
--B  High Density and High Rise Flats
--C  Settled Asians
--D  Urban Elites
--E  City Vibe
--F  London Life-Cycle
--G  Multi-Ethnic Suburbs
--H  Ageing City Fringe

SELECT
    supgrp_cd,
    count(geom)
FROM greater_london_loac
GROUP BY
    supgrp_cd
```

```
ORDER BY
  supgrp_cd;
```

Table 16: .

supgrp_cd	count
A	3241
B	3138
C	2913
D	2376
E	3523
F	3224
G	3709
H	2929

To evaluate the road network with polygons cinemas

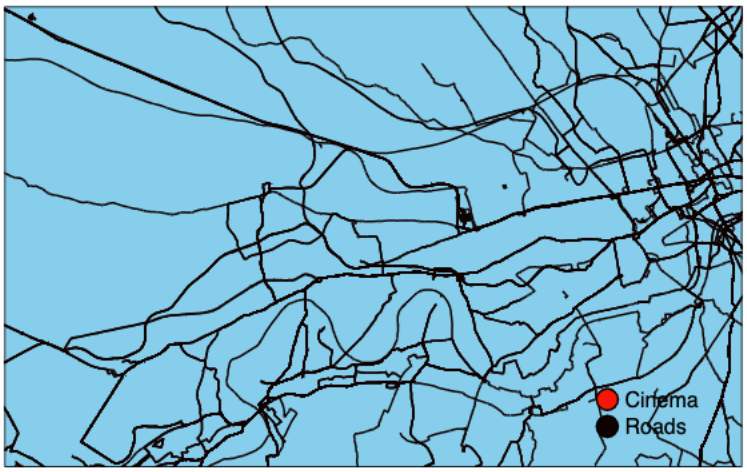
```
# Retrieve intersecting lines

intersecting_lines <- st_read(
  pgsql_conn1,
  query = "
   SELECT ply_p.name,line_l.osm_id,line_l.name,line_l.geom
      FROM greater_london_osm_polygon AS ply_p
INNER JOIN greater_london_osm_line as line_l
        ON ST_Intersects(
              ST_Buffer(
                  ST_Transform(ply_p.geom,27700),
                  10
               ),
                 ST_Transform(line_l.geom,27700)
              )
      WHERE ply_p.amenity = 'cinema';")
```

Retrive the cinema polygons

```
# Retrive the polygons
cinema_polygons <- st_read(
  pgsql_conn1,
  query = "
   SELECT *
     FROM greater_london_osm_polygon
     WHERE name = 'BFI IMAX'
        OR name = 'VUE Cenima'
        OR name = 'Cineworld';"
  )
```

Map

```
# Create a map
# use polygon shapes to fill
tm_shape(cinema_polygons) +
    # filled polygons
    tm_fill(col = "#f51707") +
  # Add the line shapes
  tm_shape(intersecting_lines) +
    # lines
    tm_lines(col = "#0f0100") +
  tm_layout(bg.color = "skyblue", inner.margins = c(0, .02, .02, .02)) +
  tm_add_legend(
    type = "symbol",
    labels = c("Cinema", "Roads"),
    col = c("#f51707", "#0f0100"))
```

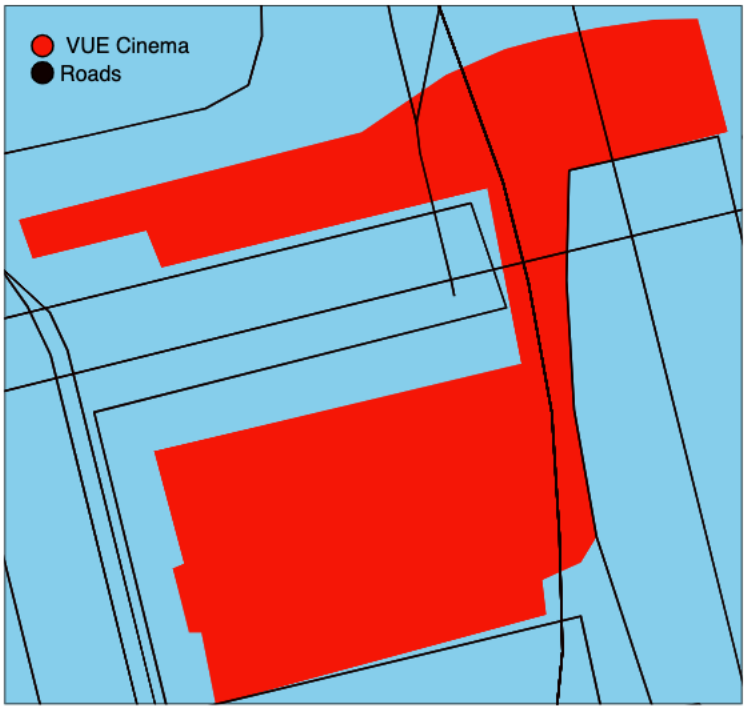


Retrive the cinema polygons

```
# Retrive the polygons
cinema_polygons1 <- st_read(
  pgsql_conn1,
  query = "
   SELECT *
     FROM greater_london_osm_polygon
     WHERE name = 'VUE Cenima';"
  )
```

Map

```
# Create a map
# use polygon shapes to fill
tm_shape(cinema_polygons1) +
    # filled polygons
    tm_fill(col = "#f51707") +
  # Add the line shapes
  tm_shape(intersecting_lines) +
    # lines
    tm_lines(col = "#0f0100") +
  tm_layout(bg.color = "skyblue", inner.margins = c(0, .02, .02, .02)) +
  tm_add_legend(
    type = "symbol",
    labels = c(" VUE Cinema", "Roads"),
    col = c("#f51707", "#0f0100"))
```



Summary

Following a detailed review, the result from the table shows that the average distance between the railway station and the cinema is around 4 kilometres, with cinema values that are very far away, by commuting from a railway station that is inaccessible to commuters. It was determined that travelling by train was not feasible, so for both point and polygon tables, we would use the bus stop and bus station located near the cinema halls. We use 2000 metres as a buffer range, as recommended by the UK government in their 2001 revision of Planning policy guidance 13: Transport (PPG13), which includes advice on appropriate walking distance.

Following the intersection, we discovered that the majority of the cinemas are within walking distance of the

bus stop or are within walking distance of the bus stop. As a result, taking the bus is a very viable choice for commuting. BFI MAX is the most famous movie theatre among tourists. The road network surrounding the cinema hall will be covered next. As a result, London has a well-developed road network that connects all of the city's movie theatres. The average Londoner is evenly distributed in the greater London area. The number of multi-ethnic suburbs is high, while the number of settled Asians is low in comparison to other races. Despite the fact that everyone in the community has access to a good road network, they can easily commute around the city.

Close connection

Connection closed.