SUBBULAKSHMI NATARAJAN

ANALYZING PUBLIC TRANSPORT ACCESSIBILITY AND DISTRIBUTION IN METROPOLITAN MELBOURNE: A POST-PANDEMIC PERSPECTIVE

STUDENT ID :34069178

Table of Contents

1. Introduction

As Victoria adjusts to post-pandemic conditions, the need for accessible public transit has increased due to the shift from remote to in-office work. Reliable public transportation is crucial in Melbourne because of the city's increased daily commuting population. The goal of Public Transport Victoria (PTV), which oversees all transportation services in Victoria, is to make travel around the state simple and efficient. Resuming and growing services after COVID-19 is part of PTV's mission to accommodate the changing

This report addresses key questions:

- How many public transport stops are there in Melbourne for each vehicle type?
- Which routes have the most stops?
- What does the typical stop count for each vehicle type?
- How accessible is public transport during peak hours?

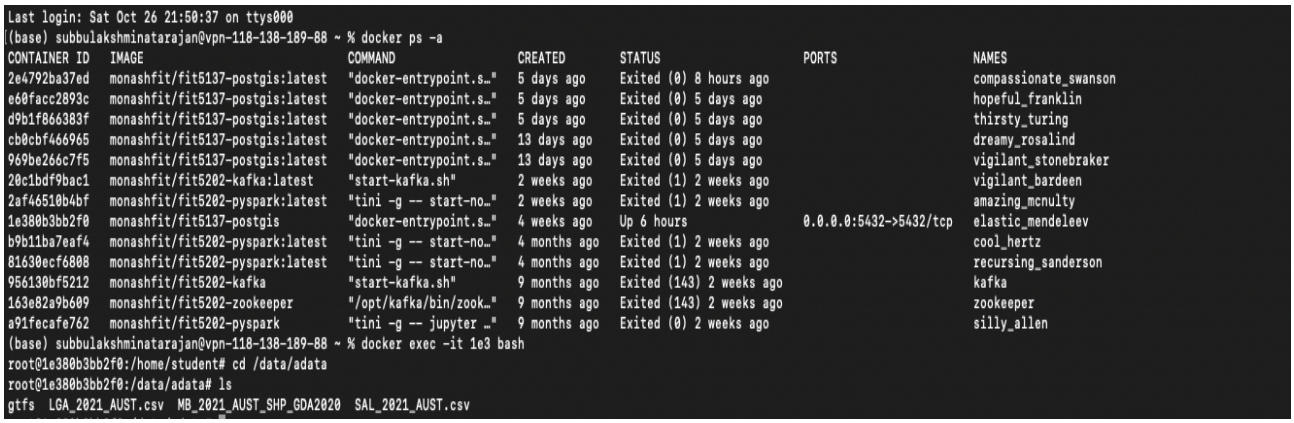
2. Methodology

Data Overview

The ABS Mesh Blocks data from the Australian Statistical Geography Standard (ASGS), which provides geographic context for analysing PTV's service coverage across various governmental levels, and the PTV General Transit Feed Specification (GTFS), which offers standardized information on routes, stops, schedules, and service frequencies of Victoria's public transit system as of the March 17, 2023, release, are the main datasets used in this analysis.

2.1 Data Restoration

To restore the data, first we need to locate the external data folder using your terminal. So, I used the Container terminal in Docker Desktop to perform the restoration. Here is the screenshot of what I did:



After that I have done **ls** to check each dataset where there are zip files in it. So, I have used **apt update && apt install unzip**. In the GTFS dataset, we have two zip files that are shapes and stop times, so we must unzip for doing the data processing as well as other steps in the report. Here is the screenshot of both datasets:

```
root@1e380b3bb2f0:/data/adata# cd gtfs
root@1e380b3bb2f0:/data/adata/gtfs# ls
agency.txt  calendar_dates.txt  calendar.txt  routes.txt  shapes.txt  shapes.txt.zip  stops.txt  stop_times.txt  stop_times.txt.zip  trips.txt
root@1e380b3bb2f0:/data/adata/gtfs# unzip shapes.txt.zip
Archive:  shapes.txt.zip
replace shapes.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: shapes.txt
root@1e380b3bb2f0:/data/adata/gtfs# unzip stop_times.txt.zip
Archive:  stop_times.txt.zip
replace stop_times.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: stop_times.txt
```

In the ABS dataset, we have two zip files that are MB_2021_AUST_GDA2020.shp.zip and MB_2021_AUST_GDA2020.dbf.zip. So, we must unzip for doing the data processing as well as other steps in the reports. Here is the screenshot of both datasets:

```
root@1e380b3bb2f0:/data/adata/MB_2021_AUST_SHP_GDA2020# ls
MB_2021_AUST_GDA2020.dbf      MB_2021_AUST_GDA2020.prj  MB_2021_AUST_GDA2020.shp.zip  MB_2021_AUST_GDA2020.xml
MB_2021_AUST_GDA2020.dbf.zip  MB_2021_AUST_GDA2020.shp  MB_2021_AUST_GDA2020.shx
root@1e380b3bb2f0:/data/adata/MB_2021_AUST_SHP_GDA2020# unzip MB_2021_AUST_GDA2020.dbf.zip
Archive:  MB_2021_AUST_GDA2020.dbf.zip
replace MB_2021_AUST_GDA2020.dbf? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: MB_2021_AUST_GDA2020.dbf
root@1e380b3bb2f0:/data/adata/MB_2021_AUST_SHP_GDA2020# unzip MB_2021_AUST_GDA2020.shp.zip
Archive:  MB_2021_AUST_GDA2020.shp.zip
replace MB_2021_AUST_GDA2020.shp? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
  inflating: MB_2021_AUST_GDA2020.shp
root@1e380b3bb2f0:/data/adata/MB_2021_AUST_SHP_GDA2020#
```

Used **ogr2ogr PG:"dbname=gisdb user=postgres" "/data/adata/MB_2021_AUST_SHP_GDA2020/MB_2021_AUST_SHP_GDA202 0/.shp" -nln ptv.australia -overwrite -nlt MULTIPOLYGON and ogr2ogr PG:"dbname=gisdb user=postgres" "/data/adata/MB_2021_AUST_SHP_GDA2020/MB_2021_AUST_SHP_GDA202 0/.dbf" -nln ptv.australia -overwrite -nlt MULTIPOLYGON** command to combine the shape files and attribute files for the ptv.australia file

After that we have created the tables and schema for both datasets and then proceed to the data preprocessing steps

2.2 Data Preprocessing

1. **Creating the Melbourne Metropolitan Boundary Polygon**

Following the selection of Melbourne's mesh blocks, we used the ST_Union function to aggregate the different mesh block geometries and produce a single border polygon for the metropolitan region. With this method, every smaller polygon is combined into a single, sizable polygon that depicts the entire metropolitan region of Melbourne.

We can directly evaluate public transport access unique to Melbourne's territory thanks to this boundary, which gives us a defined geographic scope for analysis and makes it possible to execute spatial joins more effectively when filtering stops within Melbourne.

2. **Adding a Geometry Column to the Stops Table**

We created a geom column and filled it with geometry points obtained from each stop's latitude and longitude values using the ST_MakePoint function because the Stops database originally lacked spatial data in an acceptable geometry format.

To guarantee that the geographic data corresponds precisely with other geographical layers, the coordinates were given the GDA2020 spatial reference (SRID: 7844), which is the norm for Australia. The proximity and intersection analyses that are essential to evaluating spatial accessibility may be carried out thanks to the additional geometry that makes spatial inquiries possible.

3. **Enriching Stops with Route and Vehicle Information**

We combined the Stops, Stop_Times, Trips, and Routes tables to offer more context. Each stop was linked to pertinent route data, such as route_long_name and route_short_name, using this join. Additionally, a CASE statement was used to categorise each stop by mode of transportation (such as bus, train, or tram) using the route_type column.

Analysis across many means of transportation is made possible by this enrichment, which offers a multi-dimensional view of each stop. We may address concerns regarding accessibility for every mode of transportation by linking stops to vehicle types. This is of particular importance to stakeholders who are examining Melbourne's transportation options.

4. **Filtering Enriched Stops to the Melbourne Metropolitan Area**

Lastly, we narrowed down the information to only include stops that were inside Melbourne's metropolitan area. We filtered stops that intersected with the Melbourne mesh blocks in mb2021_mel using the ST_Intersects function. This guarantees that only stops that are pertinent to the Melbourne metropolitan area are kept, enabling us to focus our investigation on the area of interest.

The dataset is aligned with the spatial boundary of interest by restricting it to Melbourne. This ensures that all analyses represent transport accessibility within Melbourne specifically, as needed by our stakeholders.

2.3 Data Analysis and Visualization

The spatial distribution and peak-hour service frequency of public transportation stations are the main topics of my investigation into the accessibility of transportation in metropolitan Melbourne. There are two primary areas of concentration here:

1. **Spatial Distribution of Transit Stops**

To determine how well various areas are covered, I look at the locations of bus stops around metropolitan Melbourne. I can determine which locations are well-served or possibly neglected by public transportation by looking at the geographic distribution and vehicle types at these stations. This phase makes use of information from the ptv.stops_melbourne table, where stops are chosen according to whether they are located inside the metropolitan limits of Melbourne.

2. **Peak Hour Service Frequency**

The ptv. peak_hour_service_frequency table illustrates the frequency of services during peak commute hours (7-9 AM and 4-6 PM), which is the second area of focus. To determine if the service satisfies high-demand accessibility requirements, this analysis counts the number of transit arrivals at each station during these crucial times. To provide a consistent image of service frequency throughout peak hours, I ascertain the hour of each service arrival and normalise the data to address temporal irregularities.

With this method, Melbourne's public transportation system can be thoroughly evaluated for both temporal and spatial accessibility, revealing any possible weaknesses and regions in need of development.

The need for accessible public transportation has surged as Melbourne shifts back to in-office work, underscoring the necessity of a dependable and effective transit system. With an emphasis on total stop availability, vehicle type distribution, route popularity, and service consistency, this analysis examines the accessibility and dispersion of public transportation in Melbourne's metropolitan area. This study

offers insights into the coverage of the transit network and identifies possible areas for improvement to better serve the needs of commuters in Melbourne using data from Public Transport Victoria (PTV) and shown in Excel and SQL.

1. **Total Number of Stops**

I evaluated Melbourne's transit system's overall stop availability. This measure aids in comprehending the fundamental reach of the transit system by offering a fundamental perspective on the network's size and infrastructure coverage. For a quick and easy reference, I utilised simple SQL Queries to show the total number of stops.

2. **Stops by Vehicle Type**

I examined how various modes of transportation contribute to Melbourne's transit accessibility by looking at stops by vehicle type (such as buses, trams, or trains). This breakdown makes it easier to see which car models are more common locations or throughout the network. The distribution of stops for each type of vehicle can be easily interpreted and compared with the help of an Excel bar chart.

3. **Top 5 Routes with the Most Stops**

I examined the top five routes with the most stops to determine which ones were the most easily accessible or regularly serviced. This draws attention to the network's most travelled routes as well as locations with high transit usage. In Excel, I created a horizontal bar chart that shows the routes with the maximum coverage by graphically ranking them from highest to lowest stop count.

4. **Average Stops per Route by Vehicle Type**

One can gain insight into route density and service consistency by mode by knowing the average number of stops per route for each type of vehicle. This metric indicates if the distribution of stops for transport types—such as buses or trains—is more consistent. Excel pie chart offer a simple way to compare different vehicle types and display the average service density for each type of route.

3. Results

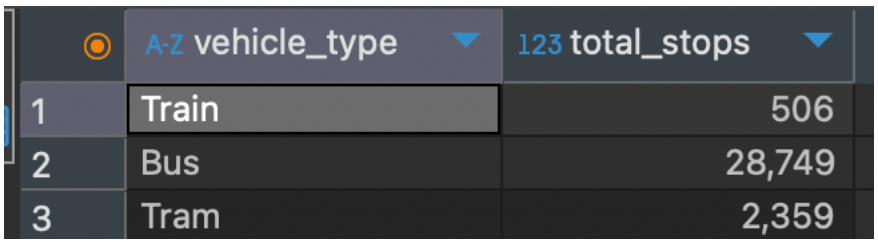
- **Total Number of Stops**

There are 31,614 stops in the Melbourne metropolitan region overall. This number illustrates the extent of the city's physical entry points for public transportation customers.

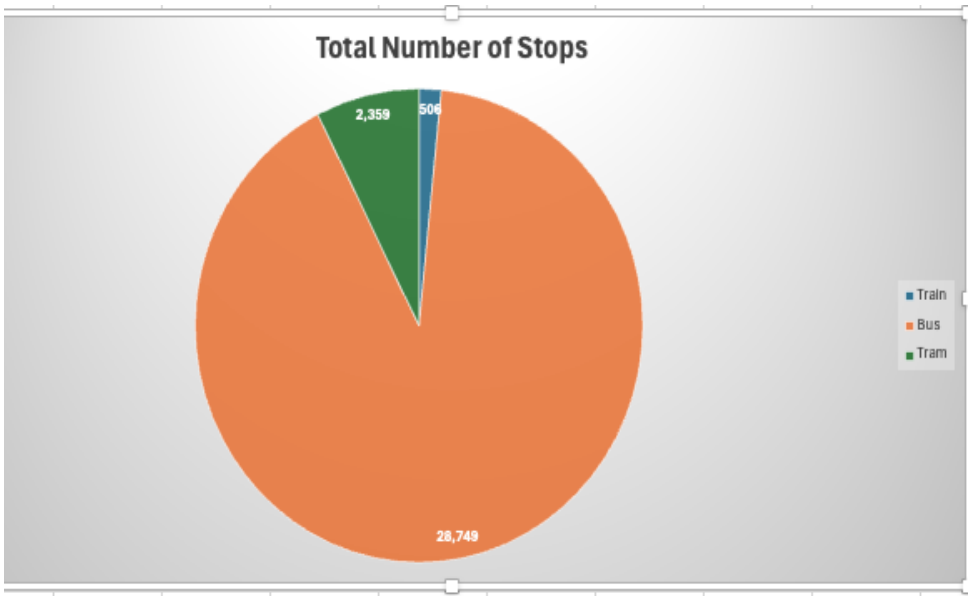
	123 total_stops
1	31,614

- **Stops by Vehicle Type**

According to the breakdown of stops by vehicle type, buses have the most stops (28,749), followed by trams (2,359) and trains (only 506). This suggests that bus networks are heavily relied upon to cover the entire metropolitan area.



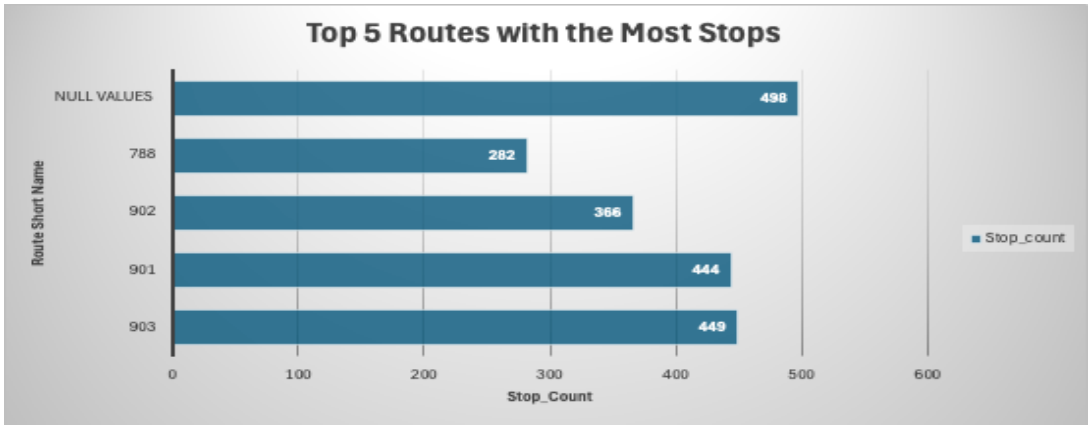
	A-Z vehicle_type	123 total_stops
1	Train	506
2	Bus	28,749
3	Tram	2,359



- **Top 5 Routes with the Most Stops**

The 903, 901, and 902 routes are among the routes with the highest number of stops, according to an analysis of the top routes. However, a sizable percentage of records have route short name null values, which could affect how clear this analysis is. To ascertain the relevance of the null values or to address data inconsistencies, more research is required.

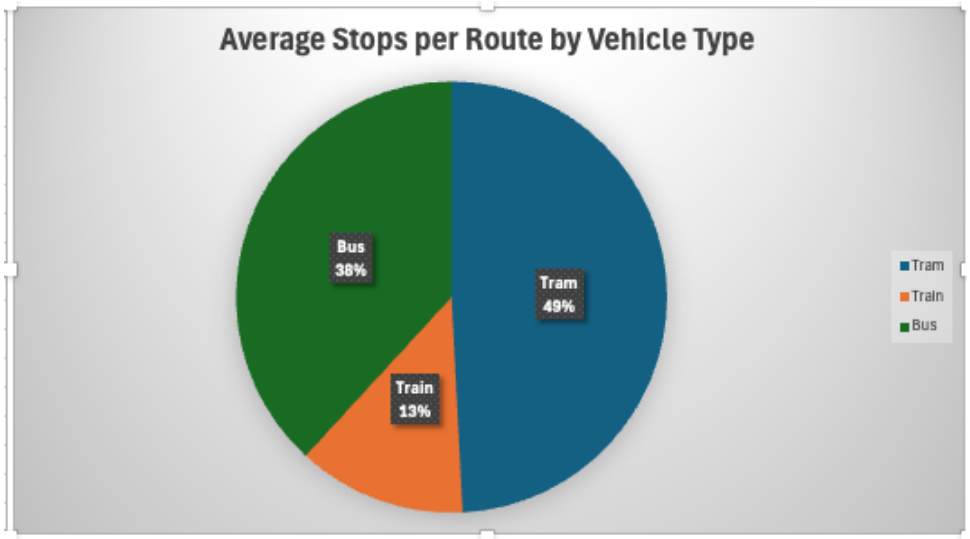
	route_short_name	stop_count
1	[NULL]	498
2	903	449
3	901	444
4	902	366
5	788	282



- **Average Stops per Route by Vehicle Type**

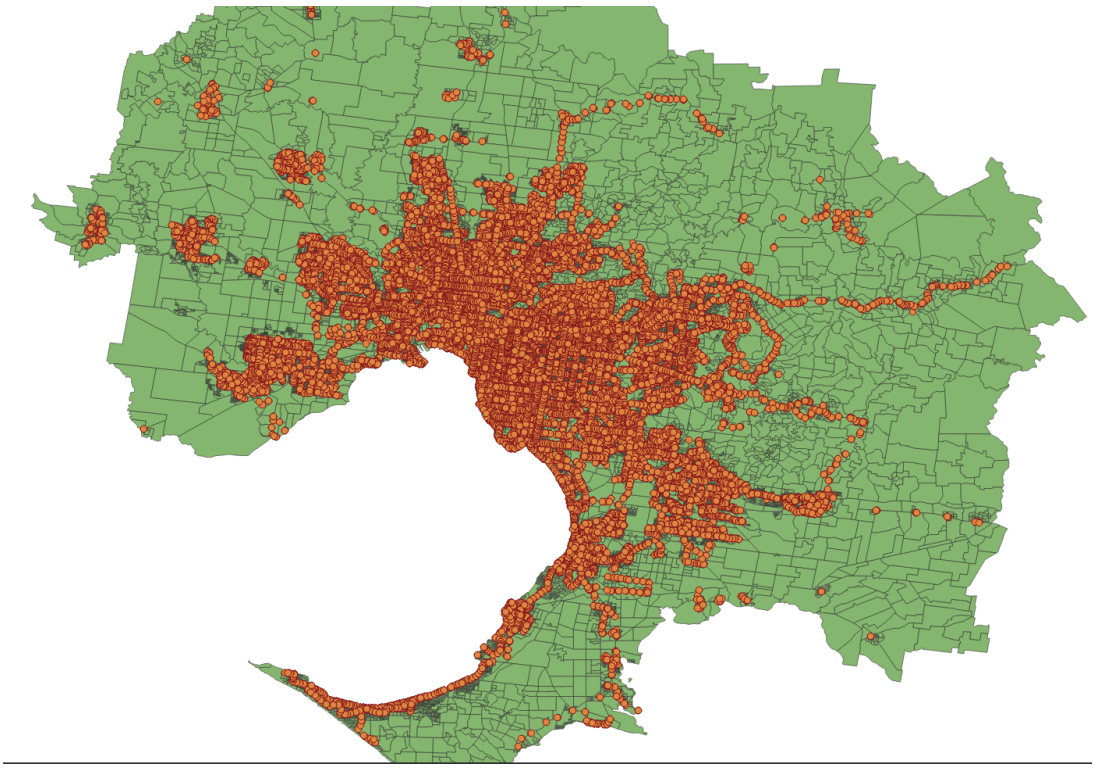
Tram routes have the most average stops (49%), followed by bus routes (38%), and train routes (13%), according to the breakdown of average stops per route. This implies that trams provide localised accessibility within their service regions because, despite their smaller numbers, they typically have more stops per route.

	vehicle_type	avg_stops
1	Tram	98.2916666667
2	Train	25.3
3	Bus	76.2572944297



- **Peak Hour Accessibility**

Accessible stops are concentrated in Melbourne's central and inner-suburban areas, where service frequency is highest during peak hours (7-9 AM and 4-6 PM), according to the map-based representation of peak hour service frequency. Indicating potential accessibility gaps for commuters living in outside suburbs, outlying districts have fewer stops and less frequent peak-hour service.



4. Discussion

The findings highlight several significant conclusions by illuminating the accessibility and distribution of public transportation stops within Melbourne's metropolitan area:

- **Dependency on Bus Networks**

Given that buses make up the bulk of stops, the bus network serves as Melbourne's main means of transportation. This extensive bus service ensures a wider reach and probably helps suburban communities and those without direct access to trams or trains. The small number of train stations, however, indicates a more centralised service and implies that trains are mostly concentrated on transit corridors with a high capacity.

- **Route Popularity and Data Gaps**

Major connectors, especially those that serve high-demand corridors like 903, 901, and 902, have the most stops. Null numbers in the data, however, are alarming and could indicate problems with the consistency or quality of the data. To provide thorough route-based insights and reliable analysis, these gaps must be filled.

- **Vehicle Type and Stop Density**

Trams often service more stops per route, according to the average stop density per vehicle type. This is in line with their function of offering frequent, closely spaced stops in inner suburbs. Trains, on the other hand, are built for quick transit over longer distances with fewer stops per route; they primarily connect important suburban centres to the metropolis.

- **Peak Hour Accessibility**

The heatmap of accessibility during peak hours shows that central Melbourne, which serves the districts with the highest commuter densities, has a notable service frequency. However, during peak hours, outlying suburbs seem underserved, which could affect commuter convenience and lengthen travel times. This discrepancy indicates that to improve accessibility generally and lessen reliance on private vehicles, peak-hour service in these locations needs to be enhanced.

Together, these results demonstrate Melbourne's dependence on its bus system, point out important routes, and indicate possible locations for service enhancement, all of which help to answer the original research concerns. Future research could concentrate on resolving problems with data quality and investigating methods to improve accessibility during peak hours in Melbourne's outer suburbs.

5. References

PostGIS Team. (n.d.). *Using PostGIS: SQL query fundamentals*. PostGIS. Retrieved October 27, 2024, from https://postgis.net/docs/using_postgis_query.html

QGIS Documentation Team. (n.d.). *Spatial databases: Spatial queries*. QGIS Training Manual. QGIS. Retrieved October 27, 2024, from https://docs.qgis.org/3.34/en/docs/training_manual/spatial_databases/spatial_queries.html

Esri. (n.d.). *Spatial SQL queries on tables with an ST_Geometry column*. ArcGIS Desktop. Retrieved October 27, 2024, from https://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/spatial-sql-queries-on-tables-with-an-st-geometry-column.htm

QGIS Documentation Team. (n.d.). *Map views*. QGIS User Manual. QGIS. Retrieved October 27, 2024, from https://docs.qgis.org/3.34/en/docs/user_manual/map_views/index.html

QGIS Tutorials and Tips. (n.d.). *Basic network analysis*. QGIS Tutorials. Retrieved October 27, 2024, from https://www.qgistutorials.com/en/docs/3/basic_network_analysis.html

QGIS. (n.d.). *Overview*. QGIS Project. Retrieved October 27, 2024, from https://qgis.org/project/overview/

6. Appendix

Task 1

Attach a screenshot of the results to include all the tables you restored in Task 1, including the number of rows for each table you restored.

```
with tbl as (select table_schema, TABLE_NAME
 from information_schema.tables
 where table_schema in ('ptv'))
 select table_schema, TABLE_NAME,
 (xpath('/row/c/text()', query_to_xml(format('select count(*) as c from %I.%I', table_schema, TABLE_NAME), FALSE, TRUE, '')))[1]::text::int AS rows_n
 from tbl
 order by table_name;
```

	table_schema	table_name	rows_n
4	ptv	lga_2021	736,572
5	ptv	mb2021_mel	59,483
6	ptv	mb_2021	368,286
7	ptv	melbourne_boundary	1
8	ptv	melbourne_metro_boundary	1
9	ptv	peak_hour_service_frequency	120,535
10	ptv	routes	3,300
11	ptv	sal_2021	736,572
12	ptv	shapes	29,272,254
13	ptv	stop_times	24,368,430
14	ptv	stops	27,821
15	ptv	stops_accessibility_peak	52,206,297
16	ptv	stops_enriched_filtered	31,614
17	ptv	stops_enrichments	41,888
18	ptv	stops_melbourne	31,614
19	ptv	stops_north_melbourne	11,191,356
20	ptv	stops_with_routes	0
21	ptv	trips	473,226

Task 2

Attach screenshots of the SQL scripts used in Task 2, including the SQL scripts for creating a table called "mb2021_mel" and screenshots of the SQL scripts you used for the remaining data processing steps.

```
-- Count all records in the `mb_2021` table where `gcc_name21` contains 'Melb'
SELECT count(*)
FROM ptv.mb_2021 m
WHERE m.gcc_name21 LIKE '%Melb%';

-- Select all records in `mb_2021` table where `gcc_name21` exactly matches 'Greater Melbourne' (case insensitive)
SELECT *
FROM ptv.mb_2021 m
WHERE lower(m.gcc_name21) = 'greater melbourne';

-- Create a new table `mb2021_mel` with data filtered for 'Greater Melbourne' in `gcc_name21`
CREATE TABLE ptv.mb2021_mel AS
SELECT *
FROM ptv.mb_2021
WHERE lower(gcc_name21) = 'greater melbourne';

-- Select all records from the newly created table `mb2021_mel`
SELECT *
FROM ptv.mb2021_mel;

-- Retrieve column names for the `mb2021_mel` table to verify table structure
SELECT column_name
FROM information_schema.columns
WHERE table_name = 'mb2021_mel';
```

```sql
-- Create a table `melbourne_metro_boundary` with a single geometry for the Greater Melbourne boundary
CREATE TABLE ptv.melbourne_metro_boundary AS
SELECT ST_Union(wkb_geometry) AS geom
FROM ptv.MB_2021
WHERE gcc_name21 = 'Greater Melbourne';

-- Select all records from the `melbourne_metro_boundary` table to check the boundary geometry
SELECT *
FROM ptv.melbourne_metro_boundary;

-- Add a new geometry column `geom` to `Stops` table to store stop coordinates in the correct SRID (7844)
ALTER TABLE ptv.Stops
ADD COLUMN geom geometry(Point, 7844);

-- Populate the `geom` column in `Stops` table by setting SRID for the point geometry of each stop
UPDATE ptv.Stops
SET geom = ST_SetSRID(ST_MakePoint(stop_lon, stop_lat), 7844);

-- Select all records from the `Stops` table to verify the new `geom` column
SELECT *
FROM ptv.stops;

-- Select all records from `routes` table for inspection
SELECT *
FROM ptv.routes;

-- Drop the `Stops_Enrichments` table if it exists to avoid duplication when creating a new version
DROP TABLE IF EXISTS ptv.Stops_Enrichments;

-- Create a new table `Stops_Enrichments` by joining `Stops`, `stop_times`, `trips`, and `routes` tables
-- and adding the `vehicle_type` column based on `route_type`
CREATE TABLE ptv.Stops_Enrichments AS
SELECT DISTINCT
    s.stop_id,
    s.stop_name,
    s.stop_lat,
    s.stop_lon,
    s.geom,
    r.route_short_name,
    r.route_long_name,
    CASE
        WHEN r.route_type = '0' THEN 'Tram'
        WHEN r.route_type = '2' THEN 'Train'
        WHEN r.route_type = '3' THEN 'Bus'
        ELSE 'Unknown'
    END AS vehicle_type
FROM ptv.Stops s
JOIN ptv.stop_times st ON s.stop_id = st.stop_id
JOIN ptv.trips t ON st.trip_id = t.service_id
JOIN ptv.routes r ON t.trip_id = r.route_id;
```

```sql
-- Create a new table `Stops_Enriched_Filtered` by filtering `Stops_Enrichments` records that intersect with Greater Melbourne geometry
CREATE TABLE ptv.Stops_Enriched_Filtered AS
SELECT
    se2.stop_id,
    se2.stop_name,
    se2.stop_lat,
    se2.stop_lon,
    se2.geom,
    se2.route_short_name,
    se2.route_long_name,
    se2.vehicle_type
FROM
    ptv.stops_enrichments se2
JOIN
    ptv.mb2021_mel m ON ST_Intersects(se2.geom, m.wkb_geometry);

-- Select all records from `Stops_Enriched_Filtered` to verify filtered data based on Greater Melbourne boundary
SELECT *
FROM ptv.stops_enriched_Filtered;
```

Task 3

The SQL script for Data analysis and visualisation

```sql
-- Count the total number of stops in the `Stops_Enriched_Filtered` table
SELECT COUNT(*) AS total_stops
FROM ptv.Stops_Enriched_Filtered;

-- Count the total number of stops for each vehicle type
SELECT vehicle_type, COUNT(*) AS total_stops
FROM ptv.Stops_Enriched_Filtered
GROUP BY vehicle_type;

-- Find the top 5 routes by the number of stops
SELECT route_short_name, COUNT(*) AS stop_count
FROM ptv.Stops_Enriched_Filtered
GROUP BY route_short_name
ORDER BY stop_count DESC          -- Order by stop count in descending order
LIMIT 5;                          -- Limit to the top 5 routes

-- Calculate the average number of stops per route for each vehicle type
SELECT vehicle_type, AVG(stop_count) AS avg_stops
FROM (
    -- Subquery to count stops for each route, grouped by vehicle type
    SELECT vehicle_type, route_short_name, COUNT(*) AS stop_count
    FROM ptv.Stops_Enriched_Filtered
    GROUP BY vehicle_type, route_short_name
) AS route_stops
GROUP BY vehicle_type;             -- Group by vehicle type to get the average stop count per route
```

```sql
-- Create a table `stops_melbourne` that contains stops within the Greater Melbourne boundary
CREATE TABLE ptv.stops_melbourne AS
SELECT
    se2.stop_id,
    se2.stop_name,
    se2.stop_lat,
    se2.stop_lon,
    se2.vehicle_type,
    se2.geom
FROM ptv.stops_enrichments se2
JOIN ptv.mb2021_mel mel ON ST_Within(se2.geom, mel.wkb_geometry);

-- Drop the table `stops_accessibility_peak` if it exists, including dependent objects
DROP TABLE IF EXISTS ptv.stops_accessibility_peak CASCADE;

-- Create a table `peak_hour_service_frequency` to store the count of services at each stop during peak hours
CREATE TABLE ptv.peak_hour_service_frequency AS
SELECT
    s.stop_id,                      -- Stop ID
    s.stop_name,                    -- Stop name
    COUNT(st.arrival_time) AS peak_service_count,  -- Count of services during peak hours

    -- Normalize `arrival_time` to handle cases where the hour value exceeds 24
    EXTRACT(HOUR FROM
        CASE
            WHEN LEFT(st.arrival_time, 2)::INT >= 24 THEN
                TIME '00:00:00' + (SUBSTRING(st.arrival_time, 4, 5) || ':00')::interval
            ELSE
                st.arrival_time::TIME
        END
    ) AS normalized_hour,           -- Extract hour from arrival time after normalization
    s.geom                          -- Stop geometry
FROM
    ptv.Stops_Enriched_Filtered s
JOIN
    ptv.stop_times st ON s.stop_id = st.stop_id
WHERE
    st.arrival_time IS NOT NULL       -- Ensure arrival time is not null
    AND (
        -- Filter for morning peak hours (7 AM to 9 AM)
        EXTRACT(HOUR FROM
            CASE
                WHEN LEFT(st.arrival_time, 2)::INT >= 24 THEN
                    TIME '00:00:00' + (SUBSTRING(st.arrival_time, 4, 5) || ':00')::interval
                ELSE
                    st.arrival_time::TIME
            END
        ) BETWEEN 7 AND 9
        -- OR for evening peak hours (4 PM to 6 PM)
        OR EXTRACT(HOUR FROM
            CASE
                WHEN LEFT(st.arrival_time, 2)::INT >= 24 THEN
                    TIME '00:00:00' + (SUBSTRING(st.arrival_time, 4, 5) || ':00')::interval
                ELSE
                    st.arrival_time::TIME
            END
        ) BETWEEN 16 AND 18
    )
GROUP BY
    s.stop_id, s.stop_name, normalized_hour, s.geom -- Group by stop details and normalized hour
ORDER BY
    s.stop_id, normalized_hour;                     -- Order by stop ID and normalized hour for readability

-- Select all records from `peak_hour_service_frequency` to verify the results
SELECT *
FROM ptv.peak_hour_service_frequency;
```