

Interim Project Documentation

1. Introduction
2. Data Sources
3. Methodology / Approach
4. Challenges
5. Analysis & Findings
6. Conclusion & Next Steps

Problem Statement

This project explores the relationship between weather patterns and healthcare demands, with a focus on how rainfall and forecast reliability may impact hospital operations. Specifically, we aim to address the following problem statements:

- Does rainfall influence emergency room admissions?
- Are weather forecasts sufficiently reliable for planning hospital resources?

For this project, we have focused the analysis on data from 2003 to 2004 for all public hospitals in Singapore. Woodlands Health has been omitted as it was operational during this time frame. Also, KKH has also been excluded as this data was not available via MOH.

Data Sources

Data	Data Type	Source
Attendances at Emergency Medicine Departments	XLXS Spreadsheet	https://www.moh.gov.sg/others/resources-and-statistics/healthcare-institution-statistics-attendances-at-emergency-medicine-departments
Historical Rainfall Data 2023	CSV File / API	https://data.gov.sg/datasets/d_36358e7cc2878156cf1e152d5b468a89/view
Historical Rainfall Data 2024	CSV File / API	https://data.gov.sg/datasets/d_a0b69d3e02576a1fd0ab673e71f83507/view
Historical 24 hour Weather Forecast 2023	CSV File / API	https://data.gov.sg/datasets/d_36358e7cc2878156cf1e152d5b468a89/view
Historical 24 hour Weather Forecast 2024	CSV File / API	https://data.gov.sg/datasets/d_0e3254694f5c7eaf77478e5fbd580f64/view
Weather Station Locations	PDF	https://www.weather.gov.sg/wp-content/uploads/2022/06/Station_Records.pdf
Rainfall Intensity Classification	PDF	https://www.nchm.gov.bt/attachment/ckfinder/userfiles/files/Rainfall%20Intensity%20classification.pdf
Hospital Locations	CSV	https://www.onemap.gov.sg/home/

Approach & Methodology

Tools

For this project we opted to use the following tools to Extract Transform and Load the weather, forecast, location and hospital data.

Tools	Use	Comments
Cloud Convert https://cloudconvert.com/xlsx-to-csv	Convert xlsx file to csv format	The Attendances at Emergency Medicine Departments data came in a xlsx spreadsheet format and we converted it to a csv format to transform the data
Python	Extract: Attendances at EMD Historical Rainfall 2023 & 2024 Weather Forecast Transform data: Attendances at EMD Weather Forecast Location mapping by region Load data to database: Historical rainfall EMD Attendances Hospital locations Hospital locations by region Weather stations by region Final database schema	<p>When working with datasets, we found that extracting the data using python was useful and hence we have opted to use it for all of our datasets.</p> <p>In addition, we also used it to transform 2 out of 3 datasets to extract additional tables such as the location mapping of the weather stations to hospital locations and regions.</p> <p>Lastly, we used python to load all of our tables into a staging database.</p>

PostgreSQL	Transform Data: Historical Rainfall 2023 & 2024 Data Cleaning Database: Staging database Final database	<p>We opted to use SQL to transform the historical rainfall data as it was more efficient than python where more complex code was needed to create our aggregation columns.</p> <p>We have also used SQL for data cleaning to ensure consistency such as to standardise data types in the date column and the hospital ids (Eg. NUH was represented in 2 different variations across the different tables)</p> <p>In addition, we are also using PostgreSQL as a database to host our data based on the designed database schema.</p>
Python Libraries	Pandas Geopandas Os Matplotlib Shapely Scikit-learn Haversine Folium Datetime Json Pdfplumber Requests Sqlalchemy Pycop2	
Tableau	Data Visualisation	

Data preparation (ETL / ELT)

For the scope of this project, we have chosen to utilise a combined approach of ETL (Extract, Transform & Load) and ELT (Extract, Load & Transform).

ELT (Extract, Load & Transform)

During the data exploration stage, we discovered that the rainfall data we obtained from NEA was much granular than what we would require for the analysis in this project. The dataset provides us with rainfall precipitation readings from 74 weather stations across Singapore in 5 min intervals for each day (See table below).

Original Data (Columns are limited to show the rainfall reading granularity)

	date text	timestamp text	station_id text	reading_value double precision	location_longitude double precision	location_latitude double precision
1	2023-12-01	2023-12-01T00:25:00+08:00	S114	0.2	103.73	1.38
2	2023-12-01	2023-12-01T00:30:00+08:00	S114	0.6	103.73	1.38
3	2023-12-01	2023-12-01T00:35:00+08:00	S114	0.4	103.73	1.38
4	2023-12-01	2023-12-01T00:40:00+08:00	S114	0.2	103.73	1.38
5	2023-12-01	2023-12-01T00:45:00+08:00	S114	0.2	103.73	1.38
6	2023-12-01	2023-12-01T00:50:00+08:00	S114	0.2	103.73	1.38
7	2023-12-01	2023-12-01T00:55:00+08:00	S114	0.4	103.73	1.38
8	2023-12-01	2023-12-01T01:00:00+08:00	S114	0.4	103.73	1.38
9	2023-12-01	2023-12-01T01:05:00+08:00	S114	0.4	103.73	1.38
10	2023-12-01	2023-12-01T01:10:00+08:00	S114	0.4	103.73	1.38

For the scope of our analysis, we only needed a **total rainfall precipitation per day** reading from the weather stations. In addition, we wanted to add in a column to determine the rainfall intensity - No Rain, Light Rain, Moderate Rain and Heavy Rain. This will be useful when comparing the forecast data.

We extracted the data for Rainfall 2023 and Rainfall 2024 via their respective CSV files and loaded them to the staging database via python source code. Then the transformations of consolidating the total rainfall per day, adding the rainfall intensity column and removing unwanted rows were performed via SQL queries.

SQL Scripts used to transform the Rainfall 2023 and Rainfall 2024 data for analysis

```
-- 1. 2023 Rainfall Data Transformation
CREATE TABLE rainfall_final2023 AS
SELECT
    date,
    station_id,
    station_name,
    location_longitude AS longitude,
    location_latitude AS latitude,
    SUM(reading_value) AS total_rainfall,
    CASE
        WHEN SUM(reading_value) > 30.01 THEN 'Heavy Rain'
        WHEN SUM(reading_value) <= 0.09 THEN 'No Rain'
        WHEN SUM(reading_value) >= 0.1 AND SUM(reading_value) <= 10.00 THEN 'Light Rain'
        ELSE 'Moderate Rain'
    END AS rain_intensity
FROM rainfall_2023
GROUP BY
    date,
    station_id,
    station_name,
    location_longitude,
    location_latitude;

--Alter date column type from text to date
ALTER TABLE rainfall_final2023
ALTER COLUMN date TYPE DATE
USING date::DATE;
```

Rainfall 2023 Table in staging database

	date date	station_id text	station_name text	longitude double precision	latitude double precision	total_rainfall double precision	rain_intensity text
1	2023-01-01	S08	Upper Thomson Road	103.8271	1.3701	0	No Rain
2	2023-01-01	S100	Woodlands Road	103.74855	1.4172	0	No Rain
3	2023-01-01	S102	Semakau Landfill	103.768	1.189	0	No Rain
4	2023-01-01	S104	Woodlands Avenue 9	103.78538	1.44387	0	No Rain
5	2023-01-01	S106	Pulau Ubin	103.9673	1.4168	0	No Rain

ETL (Extract, Transform & Load)

For all the other tables we extracted for this project, we have used the ETL (Extract, Transform & Load) approach.

Forecast Weather data

NEA provides a 24 hour weather forecast prediction that contains:

- a descriptive rainfall outlook and other measures such as windspeed, humidity and temperature;
- for 5 regions (North, South, East, West, Central) and the whole country;
- for a 24-hour “valid period” (i.e. a moving window that provides a 24-hour outlook (subdivided into 6-/12-hour “time periods”) at the time of update (6–12 times per day)).

At each update, a new record is added (for each of the three/four consecutive time periods in the current 24-hour window) of the predicted outlook for the regions and country.

A snippet of the historical 24-hour weather forecast data from NEA

Time Period Text Text	Q	Time Period Start Text	Q	Time Period End Text	Q	South Forecast Text Text	Q
(Null)	0.0%	(Null)	0.0%	(Null)	0.0%	(Null)	0.0%
Midday to 6 pm 24 Dec	0.2%	2024-12-30T18:00:00+08:00	0.2%	2024-12-31T06:00:00+08:00	0.2%	Partly Cloudy (Day)	24.0%
6 am to Midday 24 May	0.2%	2024-12-24T12:00:00+08:00	0.2%	2024-12-24T18:00:00+08:00	0.2%	Cloudy	22.9%
Midday to 6 pm 21 Jun	0.2%	2024-05-24T06:00:00+08:00	0.2%	2024-05-24T12:00:00+08:00	0.2%	Thundery Showers	22.2%
6 am to Midday 31 Dec	0.2%	2024-06-21T12:00:00+08:00	0.2%	2024-06-21T18:00:00+08:00	0.2%	Partly Cloudy (Night)	16.9%
1766 more values		1420 more values		1420 more values		11 more values	
6 am to Midday 01 Jan		2023-01-01T06:00:00+08:00		2023-01-01T12:00:00+08:00		Showers	
Midday to 6 pm 01 Jan		2023-01-01T12:00:00+08:00		2023-01-01T18:00:00+08:00		Thundery Showers	
6 pm 01 Jan to 6 am 02 Jan		2023-01-01T18:00:00+08:00		2023-01-02T06:00:00+08:00		Light Rain	
Midday to 6 pm 01 Jan		2024-01-01T12:00:00+08:00		2024-01-01T18:00:00+08:00		Thundery Showers	
6 pm 01 Jan to 6 am 02 Jan		2024-01-01T18:00:00+08:00		2024-01-02T06:00:00+08:00		Cloudy	

For the scope of our analysis, we need to narrow the data down to the **forecast of rainfall for each region in the 24-hour time periods** corresponding to each calendar day, to be able to compare it with the rainfall data. Hence we will need to aggregate the forecasts from multiple consecutive time periods and updates.




In Python, we have attempted to simplify this transformation by collapsing the predictions to a single binary distinction:

- Rain (a function returns True if '_forecast_text' contains 'Rain'|'Showers')
- No Rain (all other forecast_text, e.g. 'Cloudy', 'Fair', etc. to return as False)

Also, a key thing to note is that the rainfall forecast is represented as descriptive text rather than numerical data. As discussed in the rainfall data, we have attempted to add a rainfall intensity column with descriptive text values to match the forecast data for analysis.

After transformation, each aggregated record would thus contain the date, region and aggregated forecast (True / False) as seen in the table below.

Forecast of rainfall data in staging database

	date 	region 	rain_forecasted  boolean
1	2022-01-01	south	false
2	2023-01-01	south	true
3	2023-01-10	south	true
4	2023-01-11	south	true
5	2023-01-12	south	false
6	2023-01-13	south	true
7	2023-01-14	south	true
8	2023-01-15	south	true

Emergency Department Attendances Data

The number of attendances to the emergency department (EMD) refers to all patients who are presented at the EMD regardless of urgency of medical conditions per day.

We were able to obtain this data from the Ministry of Health (MOH) in an xlsx spreadsheet format. We first converted this to a csv file and used python to read it.

A snippet of the raw data of EMD attendances from MOH

	Date	AH	CGH	KTPH	NTFGH	NUH(A)	SGH	SKH	TTSH	WH
0	Sun, 01/01/23	64	351	286	252	257	309	333	336	NaN
1	Mon, 02/01/23	61	386	326	314	334	342	346	370	NaN
2	Tue, 03/01/23	76	436	401	364	352	343	397	422	NaN
3	Wed, 04/01/23	74	354	311	330	286	305	327	361	NaN
4	Thu, 05/01/23	61	373	335	320	309	337	351	366	NaN

Upon reviewing the data, we transformed the data using python by:

- removing the day from the date column to only show the date;
- convert it to a vertical table with columns date, hospital_id, hospital_name and attendance values.

To do this, we split the string in the date column and retained the date only, followed by using the stack method from the pandas dataframe.

The outcome results in a table that is now compatible to be used for further analysis:

	date date	hospital_id text	hospital_name text	attendance double precision
1	2023-01-01	AH	Alexandra Hospital	64
2	2023-01-01	CGH	Changi General Hospital	351
3	2023-01-01	KTPH	Khoo Teck Puat Hospital	286
4	2023-01-01	NTFGH	Ng Teng Fong General Hospital	252
5	2023-01-01	NUH(A)	National University Hospital	257

Locations Data (Derived Tables)

Thus far, we have shown how we have extracted and transformed the raw data from NEA and MOH. However, to do a proper analysis we will need to use the rainfall data, forecast data and hospital locations to map:

- Hospitals by region
- the weather stations measuring rainfall to the nearest hospital;
- the weather stations by region (to correspond to forecast data)

Hospital by region Data

We used the latitude and longitude coordinates of the hospitals' locations in Singapore to do a proper geospatial mapping to the nearest weather stations and region in which they are located.

For this, we used the GeoDataFrame from geopandas library and kmeans from the scikit-learn library to create the clusters of hospitals by region.

	hospital_id text	region text	latitude double precision	longitude double precision
1	KTPH	Central	103.8385789	1.424081
2	SKH	Central	103.8931641	1.3943931
3	TTSH	Central	103.845694	1.321368
4	CGH	East	103.9494655	1.3408259
5	AH	South	103.8001809	1.2854811
6	NUH	South	103.7836867	1.2944203
7	SGH	South	103.8355418	1.2796438
8	NTFGH	West	103.7454484	1.3336062

Weather stations to nearest hospital data

Obtaining the weather stations in proximity to the hospitals was easier as both of them have clear longitude and latitude coordinates. We used the hospital location data and the weather stations location (from rainfall data), and ran them through a define function for proximity using the haversine python library.

A snippet of weather stations to the nearest hospital table

	hospital_id text	station_id text
1	AH	S77
2	AH	S120
3	AH	S223
4	AH	S226
5	AH	S102
6	CGH	S107
7	CGH	S113
8	CGH	S94
9	CGH	S207
10	CGH	S208

Weather stations by region data

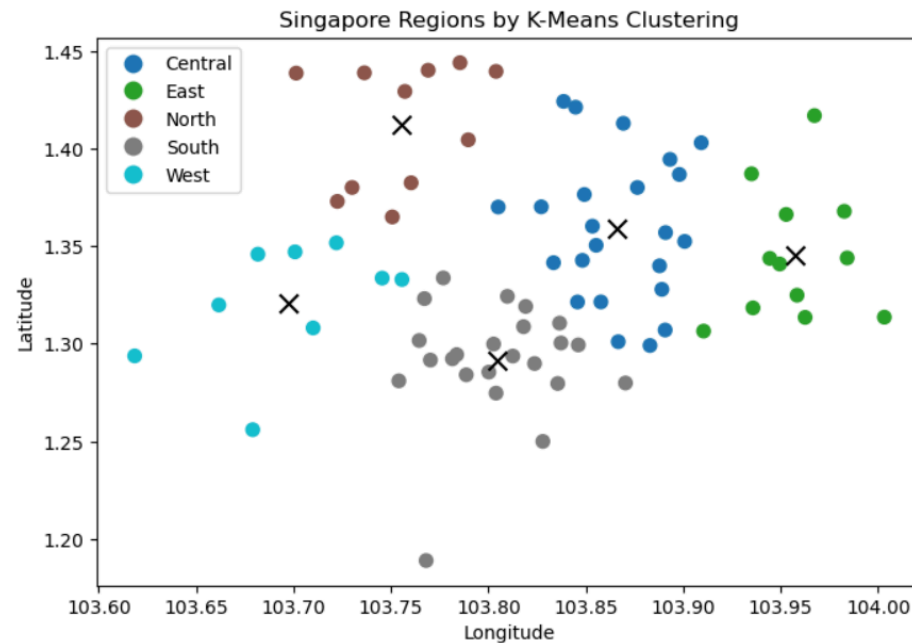
Next we wanted to group the weather stations by region to correspond them to the forecast data.

For this, we used the GeoDataFrame from geopandas library and kmeans from the scikit-learn library to create the clusters of weather stations by region.

A snippet of weather stations by region table

	region text	station_id text	longitude double precision	latitude double precision
1	Central	S109	103.8492	1.3764000000000003
2	Central	S215	103.88899000000002	1.32785
3	Central	S43	103.8878	1.3399
4	Central	S07	103.8334	1.3415
5	Central	S119	103.8666	1.30105

Dot plot of weather stations and hospitals by regions



Staging Database

Hence, our staging database now has the following tables:

Historical Rainfall 2023

rainfall_2023
date
station_id
station_name
longitude
latitude
total_rainfall
rain_intensity

Historical Rainfall 2024

rainfall_2024
date
station_id
station_name
longitude
latitude
total_rainfall
rain_intensity

EMD Attendances

emd_data
date
hospital_id
hospital_name
attendance

Hospital by region

hospitals_region
hospital_id
region
latitude
longitude

Stations by region

stns_region
region
station_id
longitude
latitude

Stations by hospital

stns_hospital
hospital_id
station_id

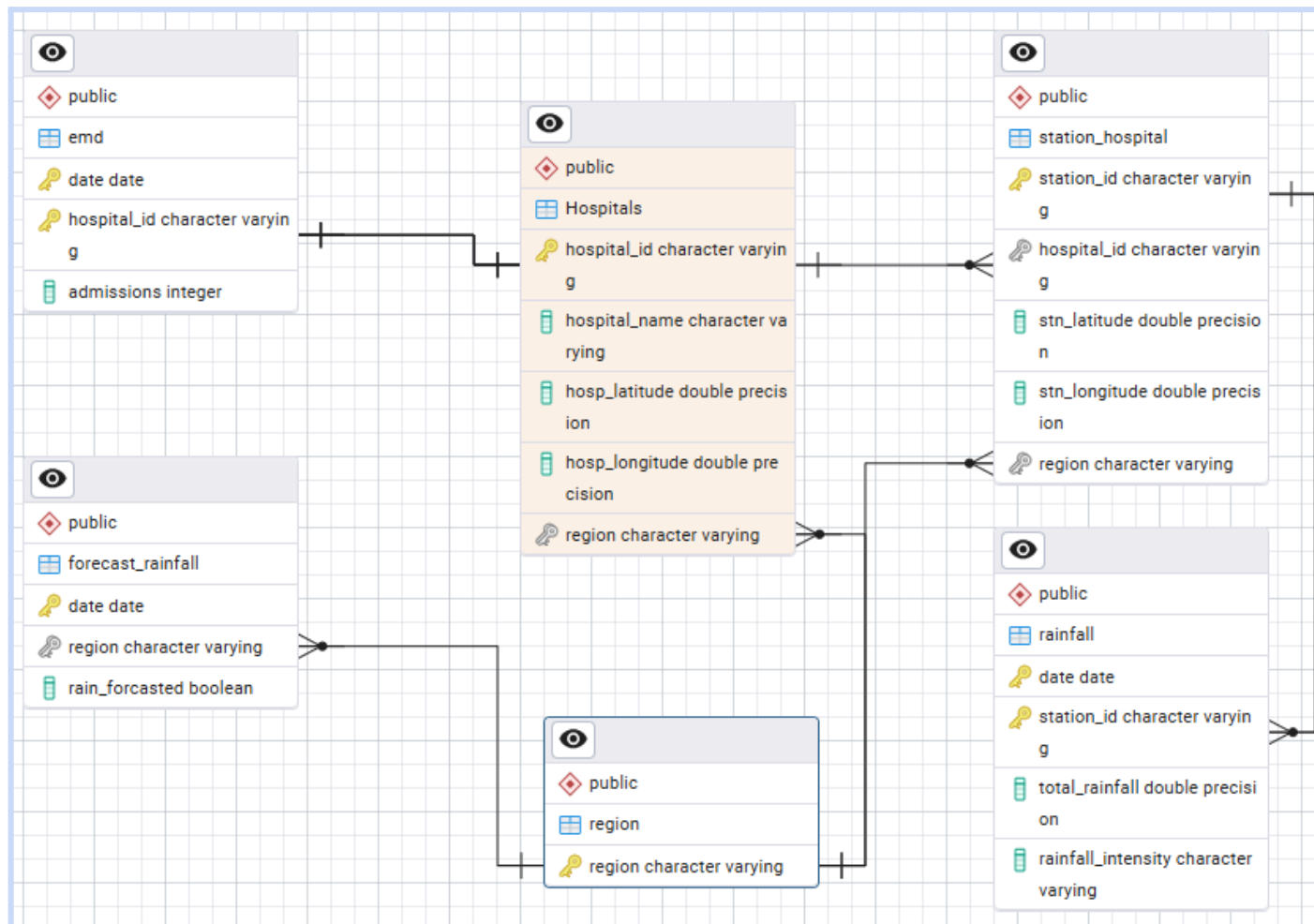
Rainfall forecast

agg_forecasts
date
region
rain_forecasted

Final Database Schema & Entity Relationship Diagram

Since our data is aggregated and needs to be optimized for OLAP queries used in data analysis, we have chosen a dimensional modeling approach with fact and dimension tables for the final database design. The fact table is shown in the middle and represented in a different color from the dimension tables.









Region is created as a separate table containing all 5 regions - south, north, east, west, central. This is necessary because the region column in the Hospitals table does not contain the south region which may be present in other data tables such as forecast_rainfall.

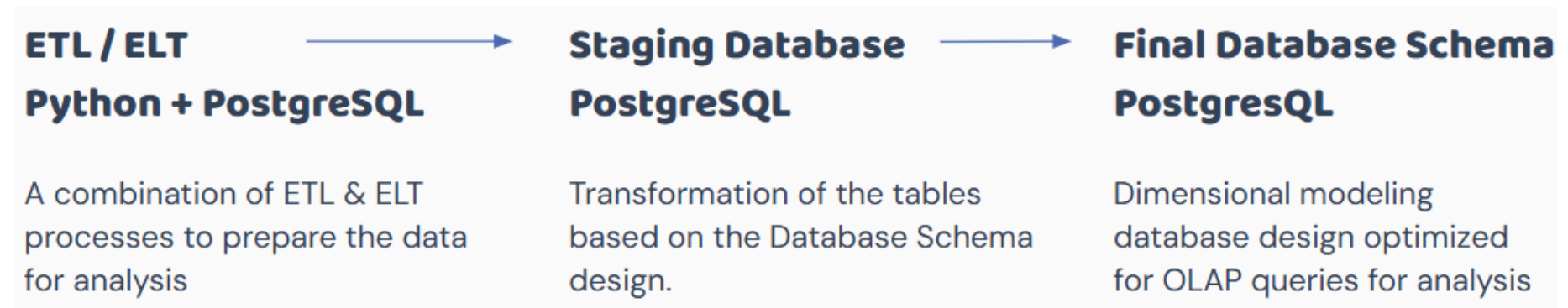


Data Flow

All python jupyter notebook and SQL scripts have been numbered accordingly for anyone to run the python source code and SQL queries to recreate this data flow for analysis. All csv files relating to each notebook have also been provided together with their respective notebooks in each of their folders.

In total, there are 7 jupyter notebooks and 2 SQL scripts.

 Name ▾
 1. Forecast Data ETL
 2. EMD Data ETL
 3. Rainfall Data ELT
 4. Location Clustering ETL
 5. Data Cleanup of Staging Database.sql
 6. Final Database Creation.ipynb
 7. Populate Final Database Tables.ipynb



Challenges

Data Size

For the rainfall data, we initially tried an ETL method to extract the data via API and perform these transformations via python.

Due to the large number of rows (6.7 million), we were limited to extracting 5000 rows at a time via API. We coded the transformation and loading to the staging database to take place one after the other. However, this took a very long time (after 5 hours and only 2 million rows were processed) and resulted in duplicate rows on the table. To prevent this, more layers of code were added as checkpoints. But the duplication could not be removed.

As this was time consuming, we opted to extract the data via its CSV file and performed the transformations in PostgreSQL directly in the staging database. This was much faster and we could complete the task in under an hour. Hence, the ELT method was more efficient for working with this data set.

Data Quality

Rainfall records and forecast records have missing date values.

Admission records do not include KKH as it was omitted from MOH report and Woodlands health has null values for 2023 & 2024 as it was not operational during this time frame.

To overcome these, we have chosen to focus on data from 2023 to 2024. Woodlands Health and KKH have been omitted for analysis. And where possible, Full Joins have been used for analysis to overcome missing date values.

Database Schema

Formulating the database schema was complex, as we needed to account for gaps in the datasets when populating them with extracted data. One issue we encountered was the presence of values in one table that were missing from another related table.

For instance, the **hospitals** table did not include the region “South,” while the **station_hospital** table did. This mismatch made it impossible to use the region column as a foreign key without omitting “South” from the **station_hospital** table.

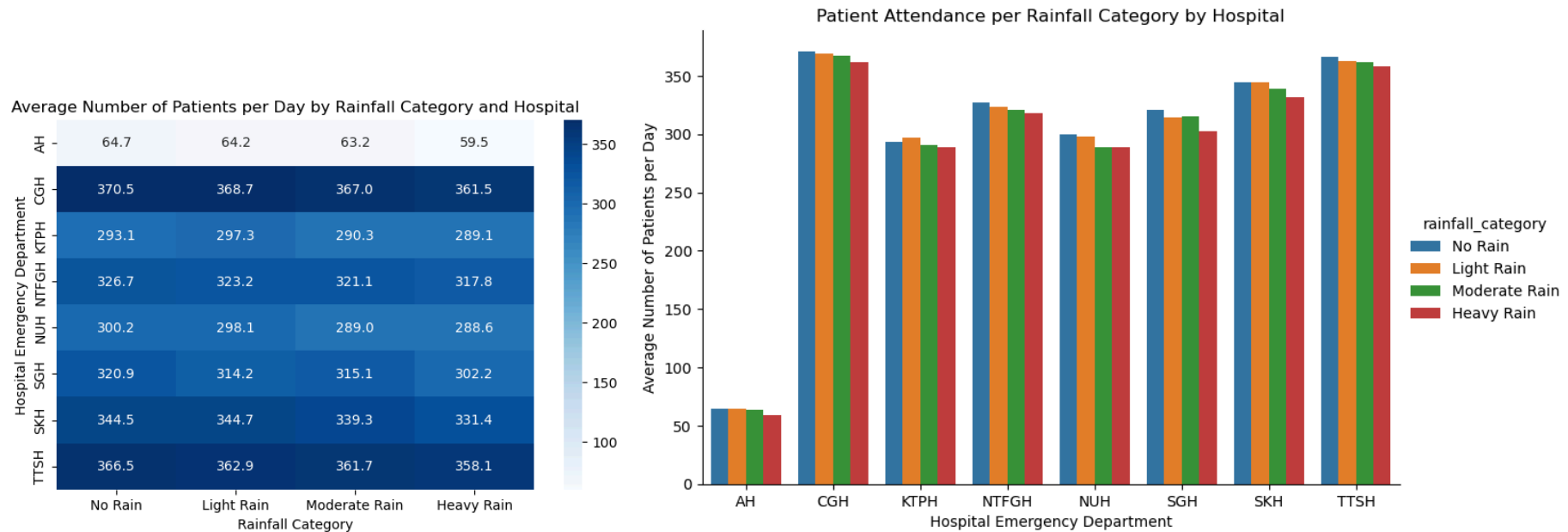
To resolve this, we created a separate table containing all regions and linked it to every table that included this column.

Findings & Analysis

Analysis Methods

We used simple aggregations to get an initial overview of the numbers and we tested if differences in admissions by rainfall intensity is statistically significant.

The heatmap showing the average number of patients per day by rainfall category and hospital reveals that this value did not vary much with higher rain intensity. With the bar chart, we are able to see this more clearly and we see that days with no or light rain have more patients in the emergency room.



To test this further, we examined whether there was a statistically significant difference in admissions between rainy and non-rainy days. Using a two-sample t-test, we obtained a p-value of approximately 0.25. Since a p-value below 0.05 is typically required to indicate statistical significance, our results suggest that the difference in admissions between the two groups is not statistically significant.

```
from scipy.stats import ttest_ind

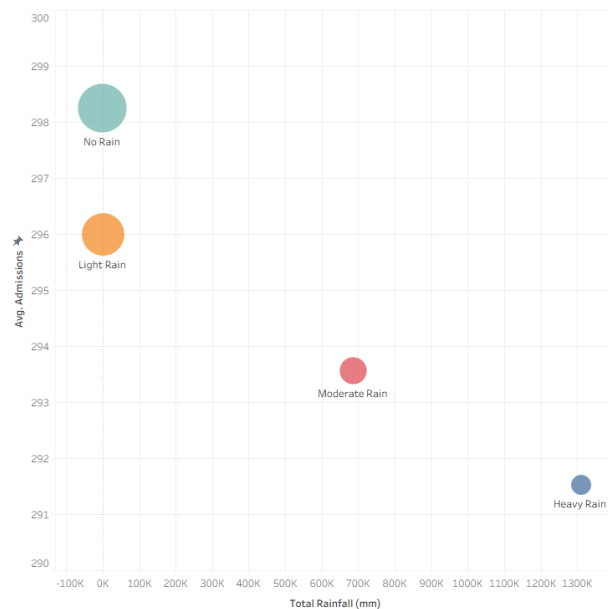
emd_rain_perday['is_rainy'] = emd_rain_perday['reading_value'] >= 0.09
emd_no_rain = emd_rain_perday[emd_rain_perday['is_rainy'] == False]['attendance']
emd_rain = emd_rain_perday[emd_rain_perday['is_rainy'] == True]['attendance']

# Welch's t-test
t_stat, p_val = ttest_ind(emd_no_rain, emd_rain, equal_var=False)

print(f"T-statistic: {t_stat:.3f}, p-value: {p_val:.5f}")
```

T-statistic: -1.151, p-value: 0.24991

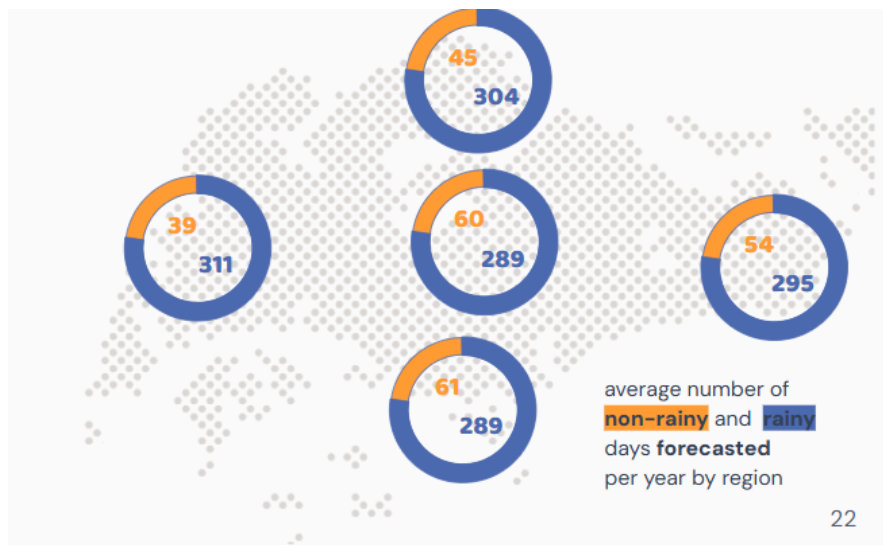
In addition, we did a simple dot plot and ran a trend line analysis to see if there was any correlation between rainfall and admissions. The R-squared value was approximately 0.0007. An R-squared value closer to 1 is required to see a correlation and our results depict that the correlation is very weak.



For forecast reliability, we found that it was 97% accurate as it did rain as predicted by the forecast.

Of the 3% of the days where it was predicted not to rain, we observe that:

- 83% were Light Rain days
- 12% were Moderate Rain days
- 4% were Heavy Rain days



```
SELECT COUNT(r.rainfall_intensity)
FROM forecast_rainfall AS fr
INNER JOIN rainfall AS r ON fr.date = r.date
WHERE rain_forecasted = 'false' AND rainfall_intensity = 'Light Rain';

SELECT COUNT(r.rainfall_intensity)
FROM forecast_rainfall AS fr
INNER JOIN rainfall AS r ON fr.date = r.date
WHERE rain_forecasted = 'false' AND rainfall_intensity = 'Moderate Rain';

SELECT COUNT(r.rainfall_intensity)
FROM forecast_rainfall AS fr
INNER JOIN rainfall AS r ON fr.date = r.date
WHERE rain_forecasted = 'false' AND rainfall_intensity = 'Heavy Rain';

SELECT COUNT(r.rainfall_intensity)
FROM forecast_rainfall AS fr
INNER JOIN rainfall AS r ON fr.date = r.date
WHERE rain_forecasted = 'false' AND rainfall_intensity <> 'No Rain';
```

Findings

Overall, we observe that Alexandra Hospital has lower admissions compared to other hospitals, regardless of rain intensity. It remains inconclusive whether rainfall is linked to an increase in ER admissions, and a hospital's location does not appear to influence how patient admissions are affected by rain. This may largely be due to the emergency department data lacking granularity, as all attendances—regardless of severity—are recorded.

In contrast, the forecasts are fairly accurate at the regional level.

Conclusion

Our analysis indicates that rainfall does not have a clear impact on ER admissions, though the evidence is not entirely conclusive. Weather forecasts provide some predictive power, but their limited reliability highlights the need for more granular analysis. Hospitals can still use forecasts to inform staffing alerts; however, they should complement these with buffer strategies to ensure preparedness. These insights may have important implications for both cost management and response times in hospital resource planning.

Next Steps

To strengthen hospital preparedness, we recommend enhancing spatial granularity with micro-weather models to better capture localized conditions. Hospitals should also explore machine learning approaches that incorporate additional parameters—**such as holidays, events, weather-related diseases, time-lagged regression, and disease incubation periods**—to more accurately predict emergency attendance surges and allocate resources proactively. Finally, integrating real-time weather feeds into hospital planning dashboards will support more comprehensive data collection and provide deeper insights for analysis.

Appendix: List of ideas Generated

Idea	Problem Statement	Sources / References
Weather Forecast Accuracy / Rainy day ER admissions correlation (Chosen Project)	<p>1. Does the daily attendance figures of patients at hospital emergency departments across Singapore correlate with whether it was raining on that day in that area?</p> <p>2. If yes, how accurate are NEA's weather forecasts?</p>	<p>The Why: https://www.nccs.gov.sg/singapores-climate-action/impact-of-climate-change-in-singapore/</p> <p>Analysing rainfall: <i>A day is considered to have "rained" if the total rainfall for that day is 0.2mm or more.]</i> https://www.weather.gov.sg/climate-climate-of-singapore/</p> <p>Rain intensity Classification https://www.nchm.gov.bt/attachment/ckfinder/userfiles/files/Rainfall%20intensity%20classification.pdf</p> <p><u>Data Sources</u></p> <p>Attendances at Emergency Medicine Departments https://www.moh.gov.sg/others/resources-and-statistics/healthcare-institution-statistics-attendances-at-emergency-medicine-departments</p> <p>Historical Rainfall Data 2023</p>

		https://data.gov.sg/datasets/d_36358e7cc2878156cf1e152d5b468a89/view Historical Rainfall Data 2024 https://data.gov.sg/datasets/d_a0b69d3e02576a1fd0ab673e71f83507/view Historical Rainfall Data 2025 https://data.gov.sg/datasets/d_6580738cdd7db79374ed3152159fbd69/view#tag/default/G%20ET/rainfall Weather Forecast Data https://data.gov.sg/datasets/d_ce2eb1e307bda31993c533285834ef2b/view
Job Portal Scraper	<p>1. I am seeking to transition to a Data Engineering role in Singapore. What are the essential skills I should have based on the job. Description?</p> <p>2. Which sectors are hiring Data engineers?</p> <ul style="list-style-type: none"> - Finance? - Ecommerce? 	<u>Data Sources</u> Careers@gov : Website scraping Linkedin : Website or API (need to check) My Careers FutureSG : Has an API (need to double check) Kaggle: https://www.kaggle.com/datasets?search=job

	<p>3. What is the salary range?</p> <p>4. Does this change with industry?</p>	<p>+scraper</p> <p>Similar projects to reference</p> <p>https://github.com/pwaaron/jobscrapers/blob/master/README.md</p>
Sentiment Analysis of Stock Market Movement	<p>1. Do sentiments from social media drive stock market movements?</p>	<p><u>Data Sources:</u></p> <p>Social Media</p> <ul style="list-style-type: none"> - X (Twitter) - reddit <p>Stock Market</p> <ul style="list-style-type: none"> - Python yfinance library - Stock market sentiment APIs <p>Similar Projects to Reference:</p> <p>https://www.kaggle.com/code/indermohanbais/news-sentiment-analysis</p>
Employee Reviews	<p>SMEs in Singapore have a bad reputation for certain workplace dysfunctions</p> <p>1. Is this purely anecdotal or is there a significant correlation between the size/type of company and employee satisfaction?</p>	<p><u>Data Sources</u></p> <p>Glassdoor: Ratings of companies on Glassdoor as a metric of employee satisfaction</p> <p>ACRA: use information from ACRA (e.g. size of company, public/private, founded locally or overseas) to categorize companies as local SMEs, foreign MNCs, etc.</p>