Impact of rainfall and weather forecasts on emergency room admissions in Singapore

Data Squirrels

```
SELECT members AS dream_team
FROM jde07
LIMIT 3;
```

Pasha, Gowry, Vee

Table of contents

01

Introduction & Project Motivation

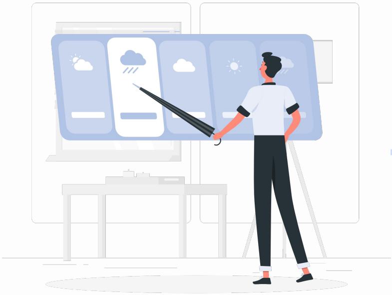
Introduction

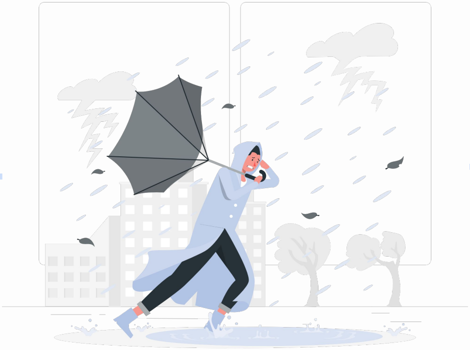
Hospitals need to anticipate surges in ER admissions.

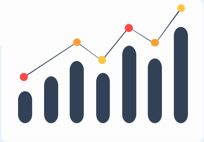
Weather conditions (e.g., heavy rainfall) may affect admission patterns.

Forecast reliability is key for proactive hospital resourcing

Cost & response time impact







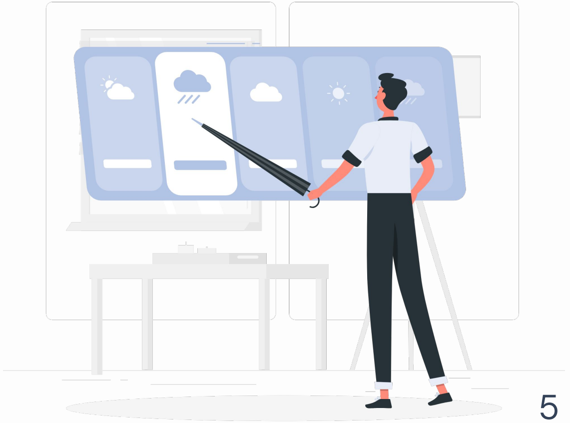
Problem Statements

01 Rainfall & Emergency Admissions

Does rainfall influence emergency room admissions?

02 Weather Forecast Reliability

Are weather forecasts sufficiently reliable for planning hospital resources?



Data Sources

National Environment Agency — Safeguard · Nurture · Cherish

Historical Rainfall 2023 & 2024

Weather Forecast

MINISTRY OF HEALTH SINGAPORE

Emergency Department Attendances

METEOROLOGICAL SERVICE SINGAPORE

Weather Station Locations

OneMap Singapore

Hospital Locations

Data Quality

Rainfall records:

Missing dates / 4 fewer weather stations in 2024 vs 2023 (70 vs 74)

Admission records:

KKH data not included in MOH report and Woodlands health has null values for 2023 & 2024 as it was not operational during this time frame

Forecast records:

Missing dates

Focus on data from **2023 to 2024**.
Woodlands Health & KKH has been omitted for analysis.
Full Joins used for analysis to overcome missing records where possible.

02

Data Extraction, Transformation and, Loading

Data Flow

ETL / ELT
Python + PostgreSQL

A combination of ETL & ELT processes to prepare the data for analysis

→

Staging Database
PostgreSQL

Transformation of the tables based on the Database Schema design.

→

Final Database Schema
PostgresQL

Dimensional modeling database design optimized for OLAP queries for analysis

All 7 python jupyter notebook and 2 SQL scripts have been numbered accordingly for anyone to run the queries and recreate this data flow in our submissions.

- 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

Transformations - EMD Data

XLSX to CSV

EMD data file

	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

Convert to a vertical table

EMD data was represented in a horizontal table

	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

Transformations - Rainfall Data

Sum of rainfall

Historical rainfall had readings for every 5 minutes per day.

Rainfall Intensity

Add a descriptive text column for the amount of rainfall

Large dataset: 6.7 million rows due to the granularity of the records (every 5 mins of reading) for each weather station. Loading via API was slow, even with chunking and sending data in batches.

Batching created duplicates when trying to aggregate the data in python before loading to the staging database.

Hence, we used the csv file to extract and performed the **transformation in postgreSQL creating an ELT pipeline.**

```sql
-- 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;
```

	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

Transformations - Forecast Data

Aggregate forecast by date

Condense 3-4 time periods within 24 hour period to 1 date

Time Period Text Text		Time Period Start Text		Time Period End Text		South Forecast Text Text	
(Null)	0.0%	(Null)	0.0%	(Null)	0.0%	(Null)	0.0%
Midday to 6 pm 7 May	0.2%	2023-02-12T06:00:00+08:00	0.2%	2023-02-12T12:00:00+08:00	0.2%	Partly Cloudy (Day)	22.6%
6 am to Midday 12 Feb	0.2%	2023-02-12T12:00:00+08:00	0.2%	2023-02-12T18:00:00+08:00	0.2%	Cloudy	21.5%
6 am to Midday 28 Dec	0.2%	2023-03-02T18:00:00+08:00	0.2%	2023-03-03T06:00:00+08:00	0.2%	Thundery Showers	20.9%
Midday to 6 pm 12 Feb	0.2%	2023-05-07T12:00:00+08:00	0.2%	2023-05-07T18:00:00+08:00	0.2%	Partly Cloudy (Night)	17.4%
1651 more values		1326 more values		1326 more values		13 more values	
6 am to Midday 1 Jan		2022-01-01T06:00:00+08:00		2022-01-01T12:00:00+08:00		Windy	
Midday to 6 pm 1 Jan		2022-01-01T12:00:00+08:00		2022-01-01T18:00:00+08:00		Windy	
6 pm 1 Jan to 6 am 2 Jan		2022-01-01T18:00:00+08:00		2022-01-02T06:00:00+08:00		Partly Cloudy (Night)	

Output forecast as binary

Use a function to collapse the predictions to a single binary distinction for each region as True or False

agg_forecasts
date
region
rain_forecasted

```python
agg_forecasts = pd.DataFrame({'fordate': sorted(df_raw_fcs['fordate'].unique())})

agg_results = []

for col in [c for c in df_raw_fcs.columns if c.endswith('_forecast_text')]:
    target_name = col.replace('_forecast_text', '')
    agg_col = (
        df_raw_fcs.groupby('fordate')[col]
        .apply(lambda x: x.str.contains('rain|showers', case=False, na=False).any())
        .reset_index(name=target_name)
    )
    agg_results.append(agg_col)

for agg_col in agg_results:
    agg_forecasts = agg_forecasts.merge(agg_col, on='fordate', how='left')

agg_forecasts['fordate'] = pd.to_datetime(agg_forecasts['fordate'])

#print(agg_forecasts)

agg_forecasts = agg_forecasts.melt(
    id_vars='fordate',
    value_vars=['south', 'north', 'east', 'central', 'west'],
    var_name='region',
    value_name='rain_forecasted'
)
```

Transformations - Location Mapping

Hospitals by region

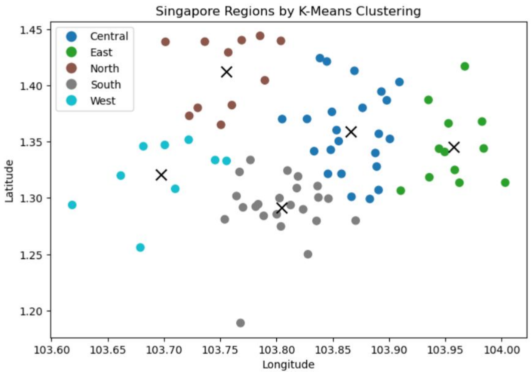
	hospital_id text	region text
1	KTPH	Central
2	SKH	Central
3	TTSH	Central
4	CGH	East
5	AH	South

Weather stations to hospitals

	hospital_id text	station_id text
1	AH	S77
2	AH	S120
3	AH	S223
4	AH	S226
5	AH	S102

Weather stations by region

	region text	station_id text
1	Central	S109
2	Central	S215
3	Central	S43
4	Central	S07
5	Central	S119



Geospatial mapping of the hospital locations to nearest weather stations in each region.

This connects data across 3 different data sets - rainfall data, hospital locations and forecast data.

Used python libraries such as **geopandas** and **kmeans** to form the clustering / proximities.

Spatial Analysis

Hospitals ↔ Weather Stations

- Haversine function to calculate the nearest hospital to each station
- group of stations near each hospital → area of rainfall near each hospital

Regions ↔ Stations, Hospitals

K-means clustering to group all stations and hospitals into the 5 regions (north, central, west, south, east) used for NEA’s weather forecasts

Staging Database

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
longtitude

Stations by region

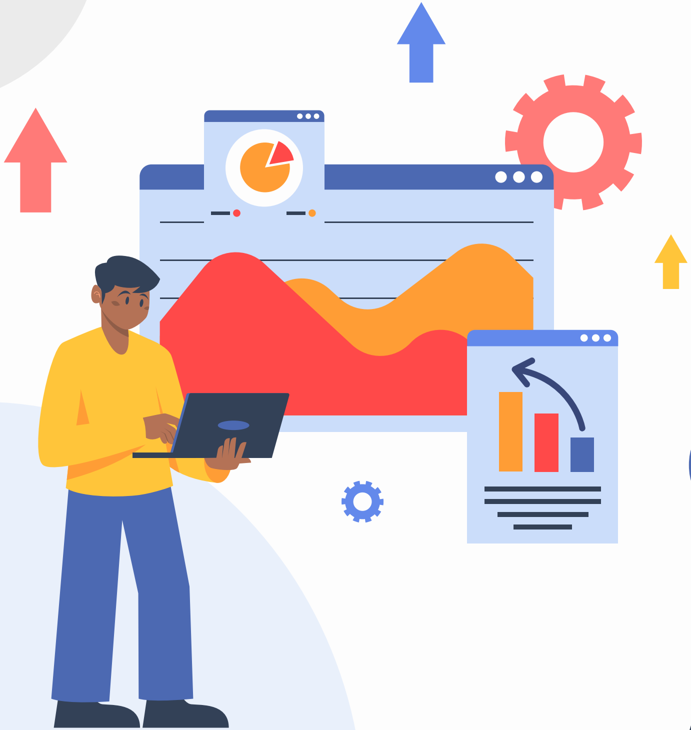
stns_region
region
station_id
longitude
latitude

Stations by hospital

stns_hospital
hospital_id
station_id

Rainfall forecast

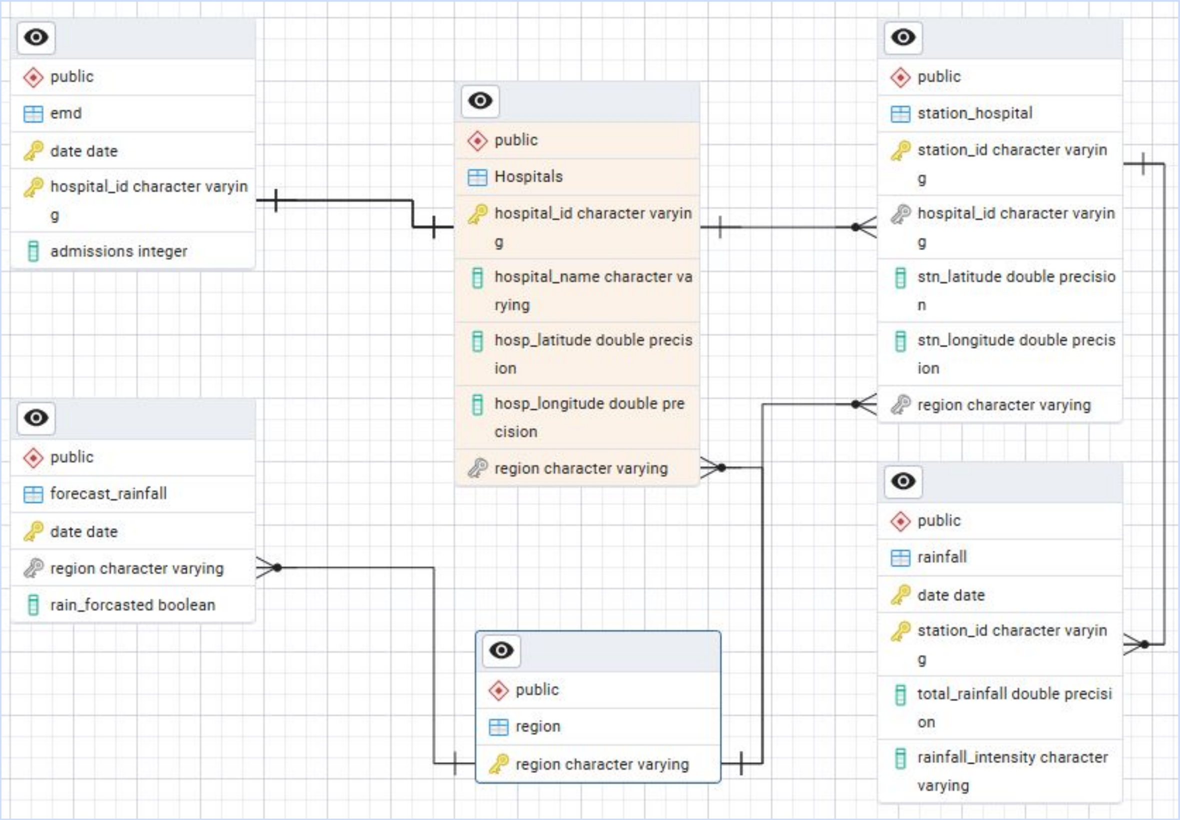
agg_forecasts
date
region
rain_forecasted



03

Database Design & Analysis

Dimensional Modelling



Analysis Methods

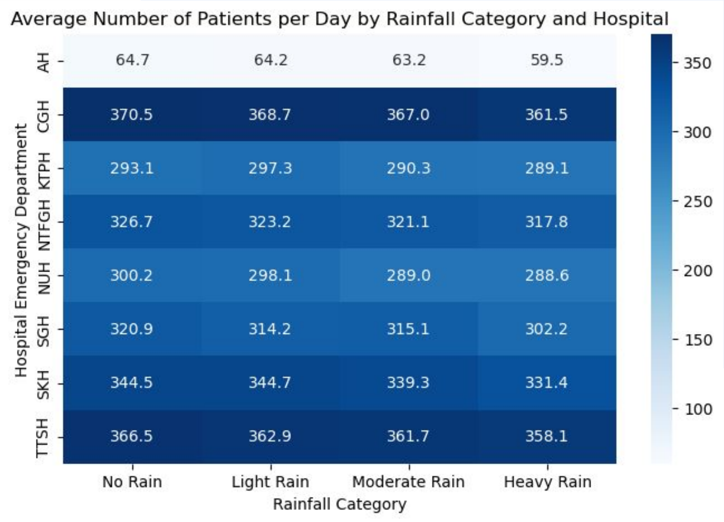
1. **Simple aggregations to get an initial overview of the numbers**

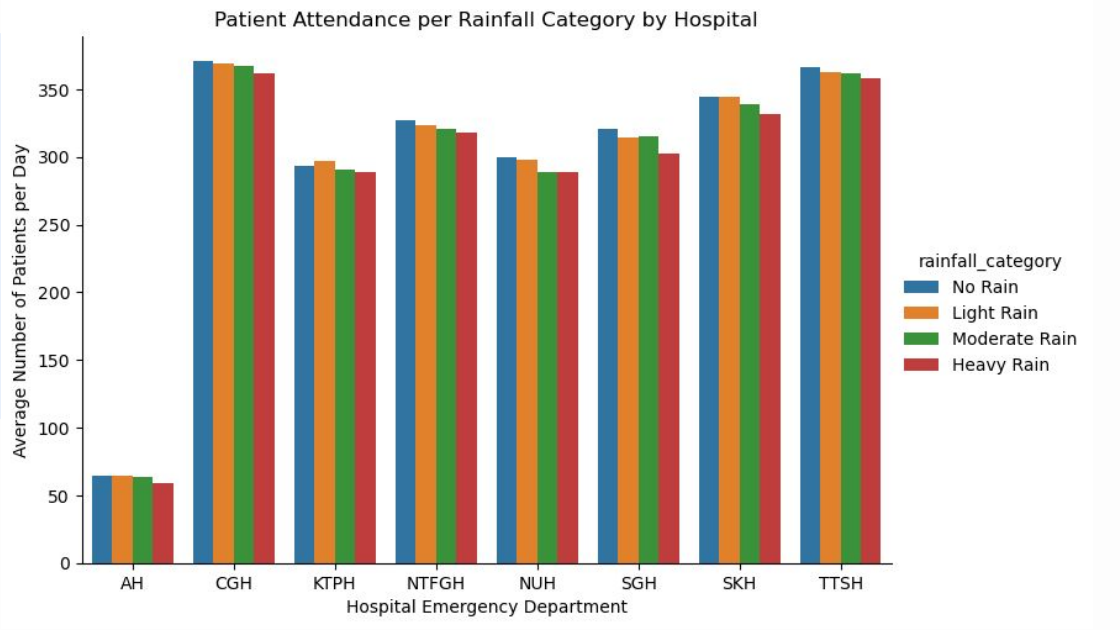
2. **Testing if differences in admissions by rainfall intensity is statistically significant**

3. **Direction for further/future analysis:**
   Could there be an effect of rainfall on hospital emergency visits on the day(s) ***following*** a rainy day?
   Run a time-lagged regression to observe potential correlation.

Rainfall Intensity vs. Patient Admissions

We observe Alexandra hospital to have lower admissions regardless of rain intensity compared to other hospitals.





Admissions on Rainy vs. Non-Rainy Days

Compare

- average number of patients admitted to the emergency dept on a rainy day
- average number of patients admitted to the emergency dept on a non-rainy day

(abstracting across different hospitals)

Is the difference statistically significant?

- perform a two-sample t-test

```
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
```

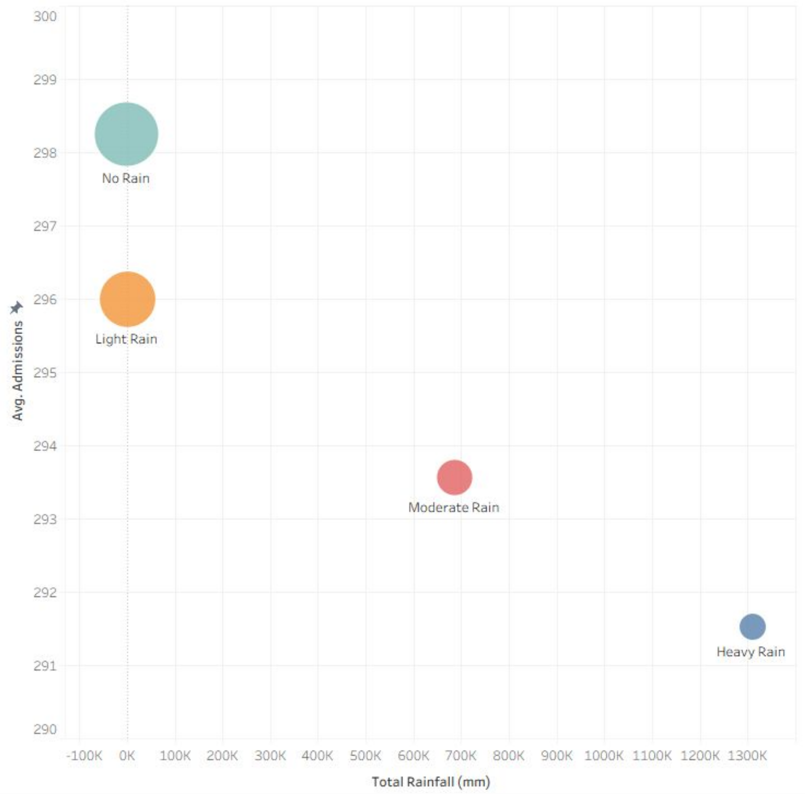
No.

Rainfall vs Admissions Correlation

Overall, during days of light and no rain, admissions were slightly higher compared to moderate or heavy rain days.

Trend line (R-squared ~ 0.0007) which shows a weak correlation.

More granular analysis is needed to gain more insights.



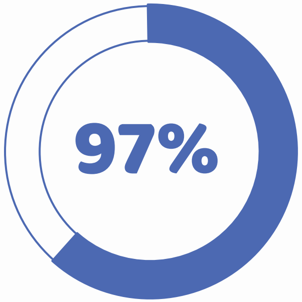
Forecast Reliability

Overall, the forecast is about **97%** accurate as it did rain as predicted by the forecast.

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

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

Preliminary Findings



Rain & ER

No conclusive Rainfall linked to a rise in ER admissions

Forecast Accuracy

~97% for region-level predictions

Weather Sensitivity

A hospital’s location does **not** influence if/how patient admissions are affected by rain

Conclusions

Findings

1 Impact on EMD

Rainfall intensity and ER admissions are inconclusive.

2 Forecast Accuracy

Forecasts provide some predictive power, but not fully reliable

3 Staffing Alerts

Hospitals may consider using other parameters for staffing alerts

4 Cost - Time

Real cost and response time implications

Next Steps

Spatial granularity

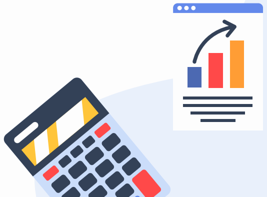
Improve spatial granularity with micro-weather models

Machine Learning

Explore machine learning models for ER surge prediction

Real-Time Weather Feeds

Integrate real-time weather feeds into hospital planning dashboards



Thanks!

Do you have any questions?

Data Squirrels

```
SELECT members AS dream_team
FROM jde07
LIMIT 3;
```

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