Group 1, Malaysia Weather Forecast

Topic: "How does temperature vary across different cities and seasons, and which weather variables (e.g., humidity, wind speed, dew point) most strongly correlate with these temperature differences?"

Group members:

- Elaine Tjandra, student id: 35008709
- Ziyue Meng, student id:36035432
- Jonathan Yek Choo Eu, student id: 36356670
- Tan Qin Tong, student id: 35862033
- Ma Chenhuan, student id: 35816880
- Koay Jiwei, student id: 36230332

Project Progress:

- Elaine Tjandra (35008709):

  30 April: created group topic

  7 May: Create individual topic

  13 May: Start coding and making graphs for my sub topic questions

  14 May: adds conclusion and revise code

- Ziyue Meng(36035432):

  30 April: created group topic

  7 May: Create individual topic

  8 May: Start coding for the sub topic

  13 May: Add Background information and conclusion

- Jonathan(36356670):

  30 April: created group topic

  7 April: created group topic and make my own sub topic questions and divide tasks among group members

  10,11 May: Start coding for my sub topic questions

12 May: analyse and draw conclusion for graph

14 May: readjust my notebook from teacher's feedback

20 May: improve the code for graph and rewrite report's conclusion from teacher feedback again

- Tan Qin Tong(35862033):

  30 April: Created group topic

  7 May: Created my own sub topic questions

  10 May: Start coding for my sub topic questions

  14 May: Analyse background and draw conclusion for charts

- Ma Chenhuan(35816880):

  30 April: Get the group topic

  5 May: Get my sub topic

  7 May: Coding, but chicken pox trouble me a lot, do not have a lot process

  11 May: fell better and start do more coding

  14 May: finish most of the coding

- Koay Jiwei(36230332):

  30 April: Get the group topic

  7 May: Created my own sub topic questions

  10,11 May: Start coding for my sub topic questions

  14 May: finish most of the coding

Background information:

Temperature variations across cities and seasons are influenced by geography, urbanization, and weather patterns. These differences impact urban planning and public health. Temperature also interacts with humidity, wind speed, and dew point, affecting perceived heat and cooling. This study analyzes these relationships to identify key weather variables linked to temperature changes, aiding climate strategies and weather predictions.

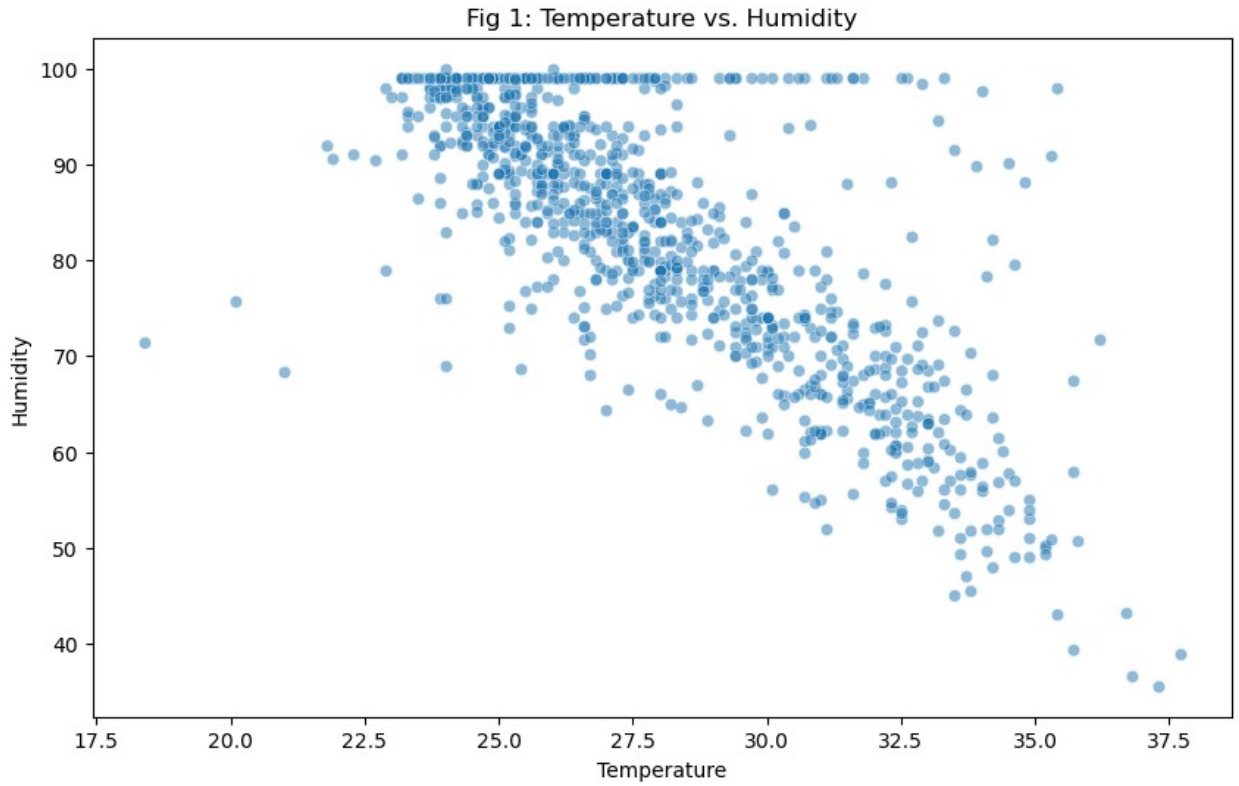
Name: Elaine Tjandra

- Question : How do temperature and humidity levels vary across different regions in Malaysia, and what is the relationship between them?
- Variables : Temperature, humidity

- Chart type : Scatter Plot - Scatter plot between temperature and humidity

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('malaysia_weather_data.csv')
df = df.sample(frac=0.02)

#check for missing values
print(df[['temperature', 'humidity']].isnull().sum())

#remove rows with missing values
df.dropna(subset=['temperature', 'humidity'], inplace=True)

#creating scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='temperature', y='humidity', alpha=0.5)
plt.title("Fig 1: Temperature vs. Humidity")
plt.xlabel("Temperature")
plt.ylabel("Humidity")

plt.show()
```

```
temperature    190
humidity       188
dtype: int64
```



Conclusion:

The graph belows demonstrates the scatter plot between two variables, temperature and humidity. It can be observed that the plots cluster in a diagonal line, which demonstrates that there is a strong negative correlation between temperature and humidity. This implies that when temperature increases, the humidity tends to decrease.

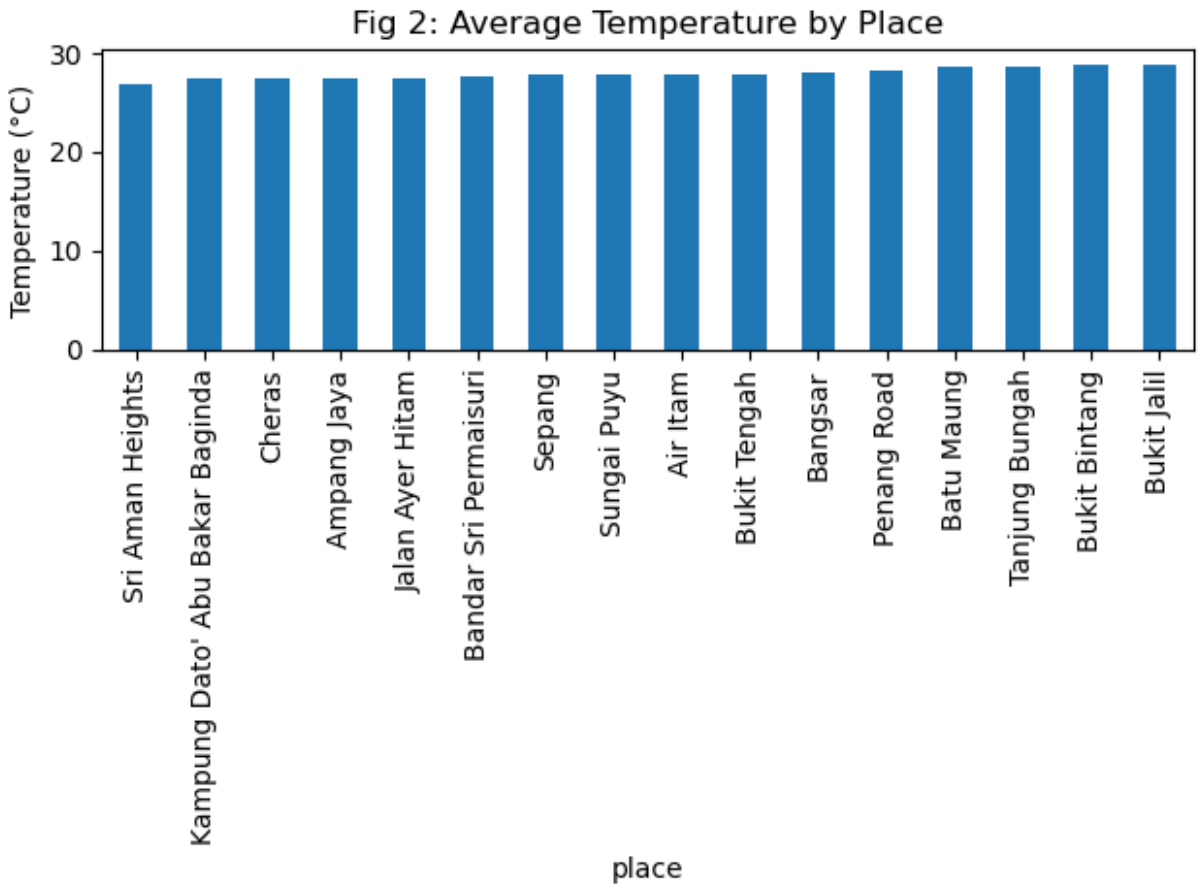
Name : Elaine Tjandra

- Question : How do temperature and humidity levels vary across different regions in Malaysia, and what is the relationship between them?
- Variables : Temperature, place
- Chart type : Bar Graph - bar graph of the average temperature across different places

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('malaysia_weather_data.csv')

#remove empty rows
df_cleaned = df.dropna(subset=['place', 'temperature'])

#calculating average temperature
avg_temperature = df_cleaned.groupby('place')['temperature'].mean()

#sort average temperature based on increasing value
avg_temperature = avg_temperature.sort_values()

#display average result by place
print(avg_temperature)

#plots bar graph
avg_temperature.plot(kind='bar', title='Fig 2: Average Temperature by
Place', ylabel='Temperature (°C)')
plt.tight_layout()
plt.show()
```

```
place
Sri Aman Heights                    26.928173
Kampung Dato' Abu Bakar Baginda     27.376723
Cheras                              27.437904
Ampang Jaya                         27.487220
Jalan Ayer Hitam                    27.529755
Bandar Sri Permaisuri               27.558152
Sepang                              27.799789
Sungai Puyu                         27.853290
Air Itam                            27.917240
Bukit Tengah                        27.937545
Bangsar                             28.007396
Penang Road                         28.223976
Batu Maung                          28.604437
```

```
Tanjung Bungah                  28.705224
Bukit Bintang                   28.877819
Bukit Jalil                     28.886960
Name: temperature, dtype: float64
```



Conclusion:

The bar graph below shows the various average temperature across different places in Malaysia and it is displayed in an ascending manner. In can be observed that Bukit Jalil has the highest average temperature while Sri Aman Heights has the lowest average temperature.

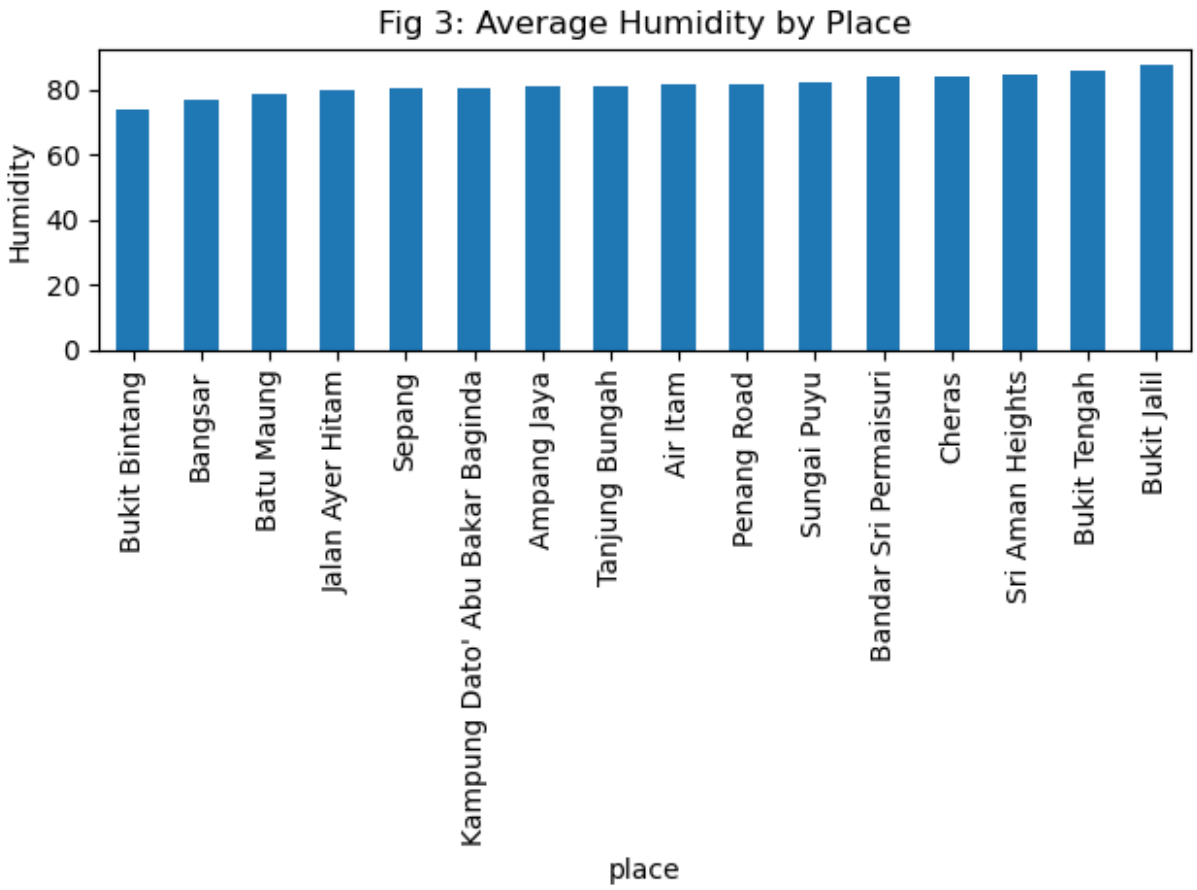
Name : Elaine Tjandra

- Question : How do temperature and humidity levels vary across different regions in Malaysia, and what is the relationship between them?
- Variables : Humidity, place
- Chart type : Bar Graph - bar graph of the average humidity across different places

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('malaysia_weather_data.csv')
```

```
#remove empty rows
df_cleaned = df.dropna(subset=['place', 'humidity'])

#calculating average humidity
avg_humidity = df_cleaned.groupby('place')['humidity'].mean()

#sort average humidity based on increasing value
avg_humidity = avg_humidity.sort_values()

#display average result by place
print(avg_humidity)

#plots bar graph
avg_humidity.plot(kind='bar', title='Fig 3: Average Humidity by
Place', ylabel='Humidity')
plt.tight_layout()
plt.show()
```

```
place
Bukit Bintang                          74.084625
Bangsar                                76.920000
Batu Maung                             78.319264
Jalan Ayer Hitam                       80.066122
Sepang                                 80.579031
Kampung Dato' Abu Bakar Baginda        80.644444
Ampang Jaya                            80.813192
Tanjung Bungah                         81.201493
Air Itam                               81.686859
Penang Road                            81.782115
Sungai Puyu                            81.931965
Bandar Sri Permaisuri                  83.711044
Cheras                                 84.239262
Sri Aman Heights                       84.479412
Bukit Tengah                           85.457702
Bukit Jalil                            87.709418
Name: humidity, dtype: float64
```



Conclusion:

The bar graph below shows the various average humidity across different places in Malaysia and it is displayed in an ascending manner. In can be observed that Bukit Bintang has the lowest average humidity while Bukit Jalil has the highest average humidity.

Name: Ziyue Meng

Subtopic Question : What are the connections and differences in temperature between Penang and Kuala Lumpur?

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline

df1 = pd.read_csv("malaysia_weather_data.csv")

df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57475 entries, 0 to 57474
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   place                 57475 non-null  object 
 1   city                  57475 non-null  object 
 2   state                 57475 non-null  object 
 3   temperature           47363 non-null  float64
 4   pressure              49285 non-null  float64
 5   dew_point             46965 non-null  float64
 6   humidity              47370 non-null  float64
 7   wind_speed            48833 non-null  float64
 8   gust                  44191 non-null  float64
 9   wind_chill            47325 non-null  float64
 10  uv_index              39177 non-null  float64
 11  precipitation_rate    46574 non-null  float64
 12  precipitation_total   46574 non-null  float64
 13  year                  57475 non-null  int64  
 14  month                 57475 non-null  object 
 15  day                   57475 non-null  int64  
 16  hour                  57475 non-null  int64  
 17  minutes               57475 non-null  int64  
 18  seconds               57475 non-null  int64  
dtypes: float64(10), int64(5), object(4)
memory usage: 8.3+ MB
```

```
# I need to convert these abbreviation for the month into integers, so I use the function: .map()
month_mapping = { 'Jan': '01', 'Feb': '02', 'Mar': '03', 'Apr': '04', 'May': '05', 'Jun': '06',
    'Jul': '07', 'Aug': '08', 'Sep': '09', 'Oct': '10', 'Nov': '11', 'Dec': '12'}
df1['month'] = df1['month'].map(month_mapping)

sort_by_place1 = df1.sort_values(['state', 'city', 'place'])
df2 = sort_by_place1.reset_index(drop=True)

state_mean = df2.groupby('state')['temperature'].mean()
state_mean
```

```
state
Kuala Lumpur    27.995131
Pulau Pinang    28.077676
Name: temperature, dtype: float64
```

```
city_mean = df2.groupby(['state','city'])['temperature'].mean()
city_mean
```

```
state         city
Kuala Lumpur  Kuala Lumpur      28.186036
```

```
              Selayang          26.928173
              Sepang            27.544417
Pulau Pinang  Balik Pulau             NaN
              Bayan Lepas       28.604437
              Bukit Mertajam    27.937545
              George Town       28.354200
              Seberang Perai    27.853290
              Timur Laut        27.917240
Name: temperature, dtype: float64
```

```
city_max = df2.groupby(['state','city'])['temperature'].max()
city_max
```

```
state         city
Kuala Lumpur  Kuala Lumpur      41.0
              Selayang          35.0
              Sepang            38.2
Pulau Pinang  Balik Pulau        NaN
              Bayan Lepas       35.0
              Bukit Mertajam    35.4
              George Town       34.9
              Seberang Perai    35.0
              Timur Laut        38.6
Name: temperature, dtype: float64
```

```
city_min = df2.groupby(['state','city'])['temperature'].min()
city_min
```

```
state         city
Kuala Lumpur  Kuala Lumpur      21.3
              Selayang          21.2
              Sepang            17.8
Pulau Pinang  Balik Pulau        NaN
              Bayan Lepas       24.0
              Bukit Mertajam    22.6
              George Town       22.8
              Seberang Perai    22.0
              Timur Laut        18.3
Name: temperature, dtype: float64
```

```
# Combine year, month, and day into a single 'date' column (commented
out for now)
df2['date'] = df2['year'].astype(str) + '-' + df2['month'] + '-' +
df2['day'].astype(str)
# Calculate the daily average temperature per state and date
daily_avg_temperature = df2.groupby(['state', 'date'])
['temperature'].mean().reset_index()
# Since grouping by date results in too many data points, the scatter
plot would become overly crowded and hard to analyze. To fix this,
I'll randomly sample 2% of the data for plotting.
```

```
sampled_df = df2.sample(frac=0.02)
# Filter the sampled data for two specific states
state1_df = sampled_df[sampled_df['state'] == 'Pulau Pinang']
state2_df = sampled_df[sampled_df['state'] == 'Kuala Lumpur']

# Find the smaller dataset size to balance samples
min_size = min(len(state1_df), len(state2_df))

# Randomly sample the same number of points from each state (for fair
comparison)
state1_sampled = state1_df.sample(n=min_size, random_state=42)
state2_sampled = state2_df.sample(n=min_size, random_state=42)


std_state1 = np.std(state1_sampled['temperature'])
std_state2 = np.std(state2_sampled['temperature'])
print(f"The standard deviation of the data in Penang: {std_state1}")
print(f"The standard deviation of the data in Kuala Lumpur:
{std_state2}")

# Create a 2x2 grid of plots (scatterplots and boxplots)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(12, 10))

# Draw the daily average temperature scatter plot of Pulau Pinang
sns.scatterplot(x='date', y='temperature', data=state1_sampled,
ax=ax1, alpha=0.7)
ax1.set_title('Fig 4: Daily Average Temperature Scatter Plot of Pulau
Pinang')
ax1.set_xlabel('Date')
ax1.set_ylabel('Average Temperature')

# Add statistical text next to the sub - plot of Pulau Pinang
stats_text1 = f'Std:{std_state1:.2f}'
ax1.text(0.05, 0.9, stats_text1, transform=ax1.transAxes, fontsize=10)

# Draw the box plot of temperature in Pulau Pinang
sns.boxplot(y='temperature', data=state1_sampled, ax=ax3)
ax3.set_title('Fig 6: Box Plot of Temperature in Pulau Pinang')
ax3.set_ylabel('Average Temperature')

# Draw the daily average temperature scatter plot of Kuala Lumpur
sns.scatterplot(x='date', y='temperature', data=state2_sampled,
ax=ax2, alpha=0.7)
ax2.set_title('Fig 5: Daily Average Temperature Scatter Plot of Kuala
Lumpur')
ax2.set_xlabel('Date')
ax2.set_ylabel('Average Temperature')

# Add statistical text next to the sub - plot of Kuala Lumpur
stats_text2 = f'Std:{std_state2:.2f}'
```

```
ax2.text(0.05, 0.9, stats_text2, transform=ax2.transAxes, fontsize=10)

# Draw the box plot of temperature in Kuala Lumpur
sns.boxplot(y='temperature', data=state2_sampled, ax=ax4)
ax4.set_title('Fig 7: Box Plot of Temperature in Kuala Lumpur')
ax4.set_ylabel('Average Temperature')

plt.show()
```

```
The standard deviation of the data in Penang: 2.809512118594799
The standard deviation of the data in Kuala Lumpur:3.134345241064917
```

Fig 4: Daily Average Temperature Scatter Plot of Pulau Pinang

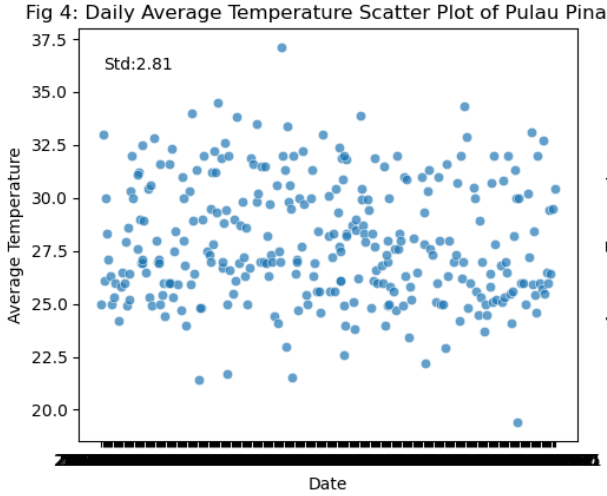


Fig 5: Daily Average Temperature Scatter Plot of Kuala Lumpur

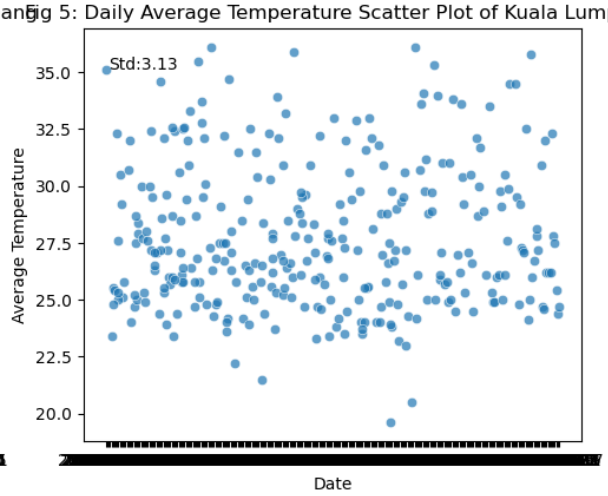


Fig 6: Box Plot of Temperature in Pulau Pinang

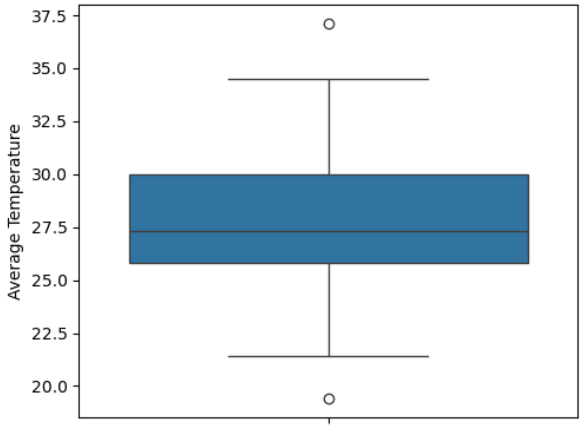
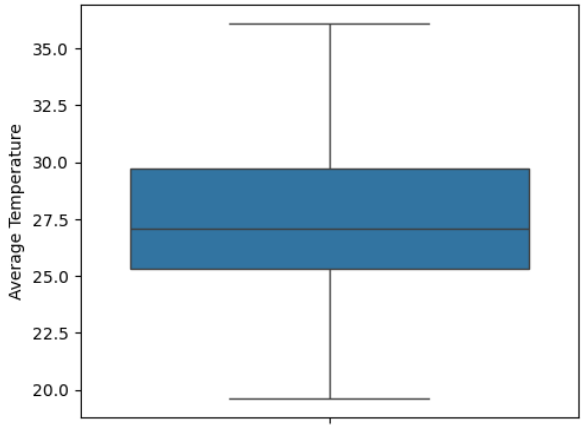


Fig 7: Box Plot of Temperature in Kuala Lumpur



Conclusion: (Ziyue Meng)

1.Temperature Variability: The standard deviation of daily average temperatures in Kuala Lumpur (3.21) is slightly higher than that in Pulau Pinang (2.90), indicating greater variability in Kuala Lumpur's temperatures.

2.Data Distribution: The box plots suggest potential differences in temperature distribution between the two locations, though further analysis (e.g., median, quartiles) is needed for detailed insights.

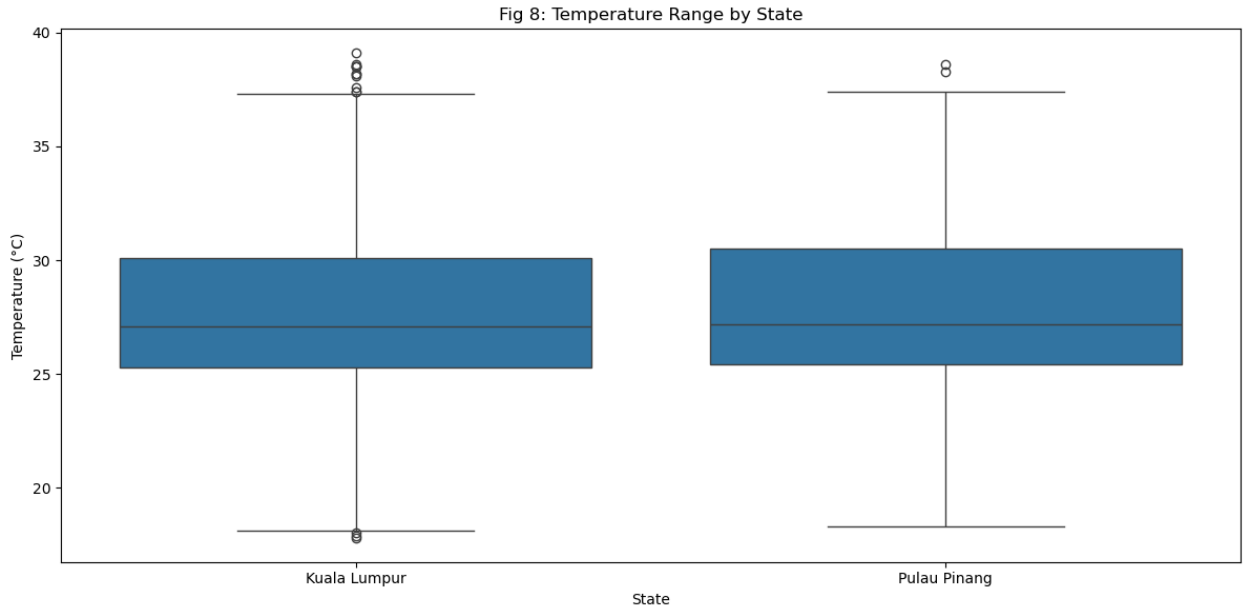
3.Visualization Effectiveness: The scatter plots and box plots effectively illustrate the distribution and dispersion of temperatures, making them suitable for comparing climate stability between the two regions.

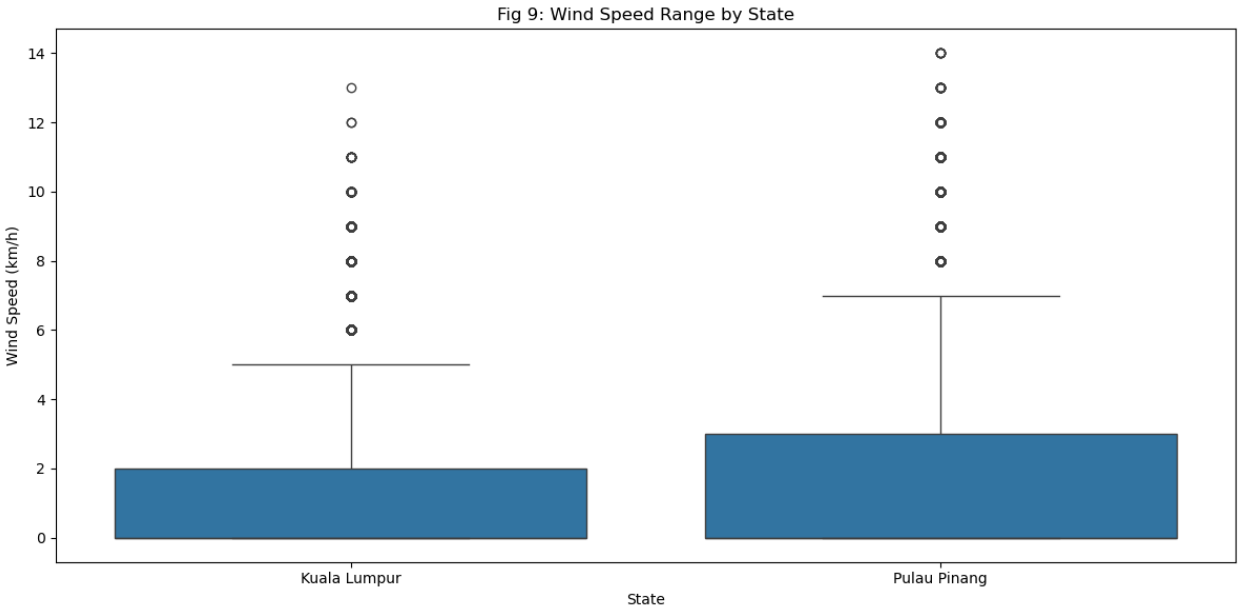
Name: Jonathan Yek Choo Eu

Subtopic Question 1: Do wind patterns (speed/gusts) explain regional temperature differences?

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Define column headers based on your data structure
headers = [
    "Station", "District", "State", "Temperature", "Pressure",
"Dew_Point", "Humidity",
    "Rainfall", "Rainfall_2", "Max_Temperature", "Wind_Speed",
"Wind_Direction",
    "Wind_Gust", "Col1", "Year", "Month", "Day", "Hour", "Min", "Col2"
]

# Load the data with headers
df = pd.read_csv('malaysia_weather_data.csv', names=headers,
header=None, dtype=str, low_memory=False)

# Convert relevant columns to numeric
for col in ['Temperature', 'Wind_Speed', 'Wind_Gust']:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Drop rows with missing essential data
df_clean = df.dropna(subset=['State', 'Temperature', 'Wind_Speed',
'Wind_Gust'])

# --- 1. Box plots: Wind speed vs. temperature ranges by state ---

plt.figure(figsize=(12, 6))
sns.boxplot(data=df_clean, x='State', y='Temperature')
plt.title('Fig 8: Temperature Range by State')
plt.xlabel('State')
plt.ylabel('Temperature (°C)')
plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(data=df_clean, x='State', y='Wind_Speed')
plt.title('Fig 9: Wind Speed Range by State')
plt.xlabel('State')
plt.ylabel('Wind Speed (km/h)')
plt.tight_layout()
plt.show()
```

```
# --- 2. Bubble chart: Avg. wind speed vs. avg. temp, bubble colour =
gust intensity ---

plt.figure(figsize=(10, 7))
scatter = plt.scatter(
    region_group['Wind_Speed'],
    region_group['Temperature'],
    c=region_group['Wind_Gust'],
    cmap='viridis',  # or 'plasma', 'coolwarm', etc.
    s=500,
    alpha=0.8,
    edgecolors='k'
)
for _, row in region_group.iterrows():
    plt.text(row['Wind_Speed'], row['Temperature'], row['State'],
fontsize=10, ha='center', va='center')
plt.xlabel('Average Wind Speed (km/h)', fontsize=12)
plt.ylabel('Average Temperature (°C)', fontsize=12)
plt.title('Fig 10: Wind Speed vs. Temperature (Color = Gust
Intensity)', fontsize=14)
cbar = plt.colorbar(scatter)
cbar.set_label('Average Wind Gust (km/h)')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```





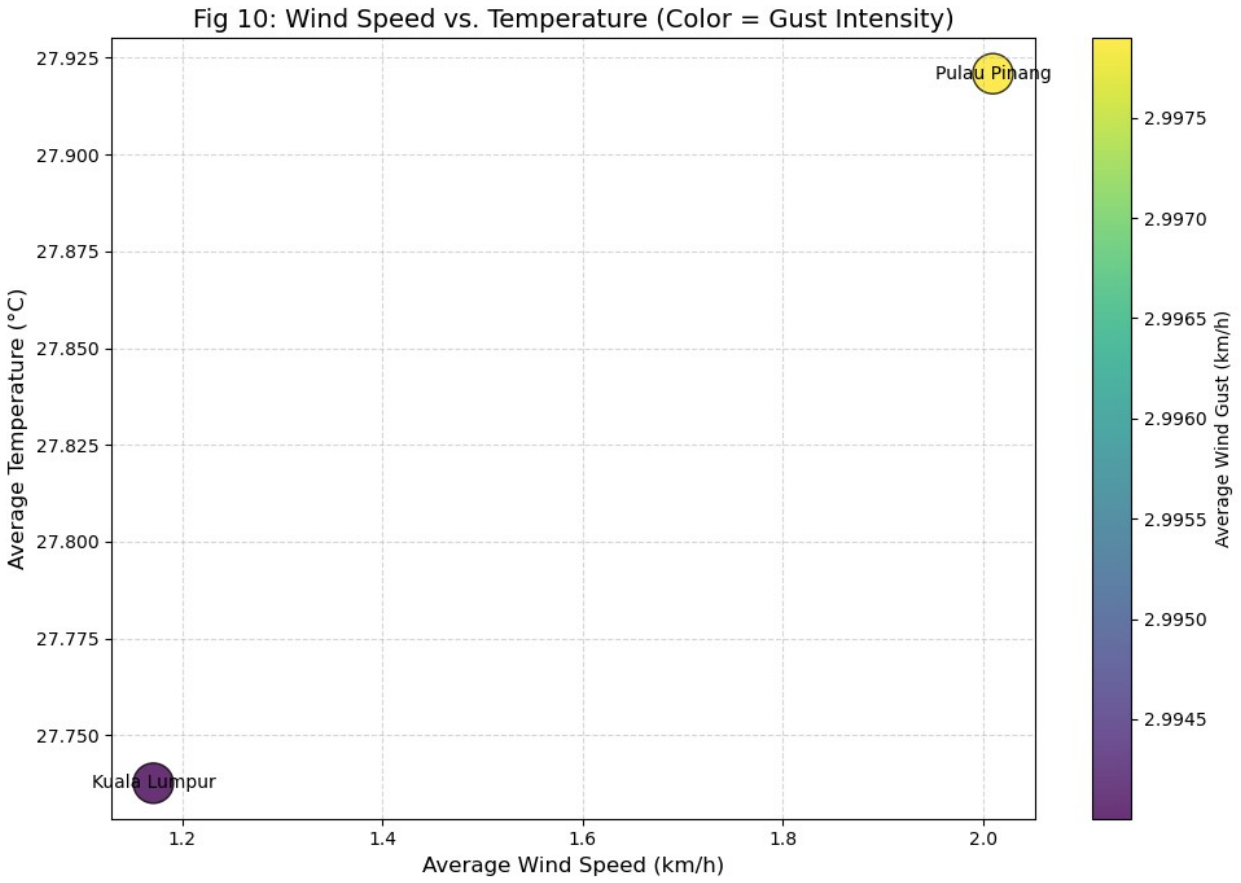


Fig 8&9: Box Plots Analysis (Temperature and Wind Speed by State)

What the plots show:

The box plots display the distribution of temperature and wind speed for each state.

Each box summarizes the median, quartiles, and outliers for the respective variable in each state.

Key observations:

Temperature:

Both Kuala Lumpur and Pulau Pinang show a similar median temperature, with some variability and occasional outliers.

The temperature range is relatively narrow, reflecting Malaysia's tropical climate.

Wind Speed:

Wind speed varies more between states than temperature.

Pulau Pinang generally exhibits slightly higher wind speeds and more variability compared to Kuala Lumpur.

Outliers in wind speed suggest occasional strong winds in certain locations or times.

Interpretation:

While temperature is fairly consistent across regions, wind speed can differ more significantly by state.

The box plots help identify which states are prone to higher or more variable wind speeds, but do not show a direct relationship between wind speed and temperature.

Fig 10: Bubble Chart Analysis (Average Wind Speed vs. Average Temperature, Bubble Colour = Gust Intensity)

What the plot shows:

Each point represents a state, plotted by its average wind speed (x-axis) and average temperature (y-axis).

The colour of the bubble reflects the average wind gust intensity for that state.

Key observations:

Most states cluster in a narrow temperature band, but wind speeds and gust intensities show more variation.

There is no strong linear relationship between average wind speed and average temperature: states with higher wind speeds do not necessarily have higher or lower temperatures.

Bubble colour (gusts) vary, but states with larger gusts are not systematically hotter or colder than others.

Interpretation:

Wind gust intensity (bubble colour) does not appear to be a primary driver of temperature differences between states.

The bubble chart visually confirms that regional temperature differences are not strongly explained by wind speed or gust intensity.

Conclusion: (Jonathan)

Box plots reveal that temperature is stable across states, while wind speed can be more variable.

Bubble chart shows no clear pattern linking wind speed or gust intensity to temperature differences.

Overall: In your dataset, wind patterns (speed/gusts) do not explain regional temperature differences in Malaysia.

Name:Tan Qin Tong

Topic: What time of day typically records the highest temperature, and is this consistent across states?

Variables used: state, temperature, hour

Appropriate chart types: line chart, heatmap

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd

df = pd.read_csv("malaysia_weather_data.csv")

temp_data = df[['state', 'hour', 'temperature']].dropna()

hourly_temp = temp_data.groupby('hour')['temperature'].mean()

plt.figure(figsize=(10, 6))
sns.lineplot(x=hourly_temp.index, y=hourly_temp.values, marker='o')
plt.title('Fig 11: Average Temperature by Hour (All Malaysia)',
fontsize=14)
plt.xlabel('Hour of the Day', fontsize=12)
plt.ylabel('Temperature (°C)', fontsize=12)
plt.xticks(range(0, 24))
plt.grid(True)
plt.tight_layout()
plt.show()

pivot_table = temp_data.groupby(['state', 'hour'])
['temperature'].mean().unstack()

plt.figure(figsize=(14, 8))
sns.heatmap(pivot_table, cmap='YlOrRd', annot=False,
cbar_kws={'label': 'Avg Temp (°C)'})
plt.title('Fig 12: Average Hourly Temperature by State', fontsize=14)
plt.xlabel('Hour of the Day', fontsize=12)
```

```
plt.ylabel('State', fontsize=12)
plt.tight_layout()
plt.show()
```

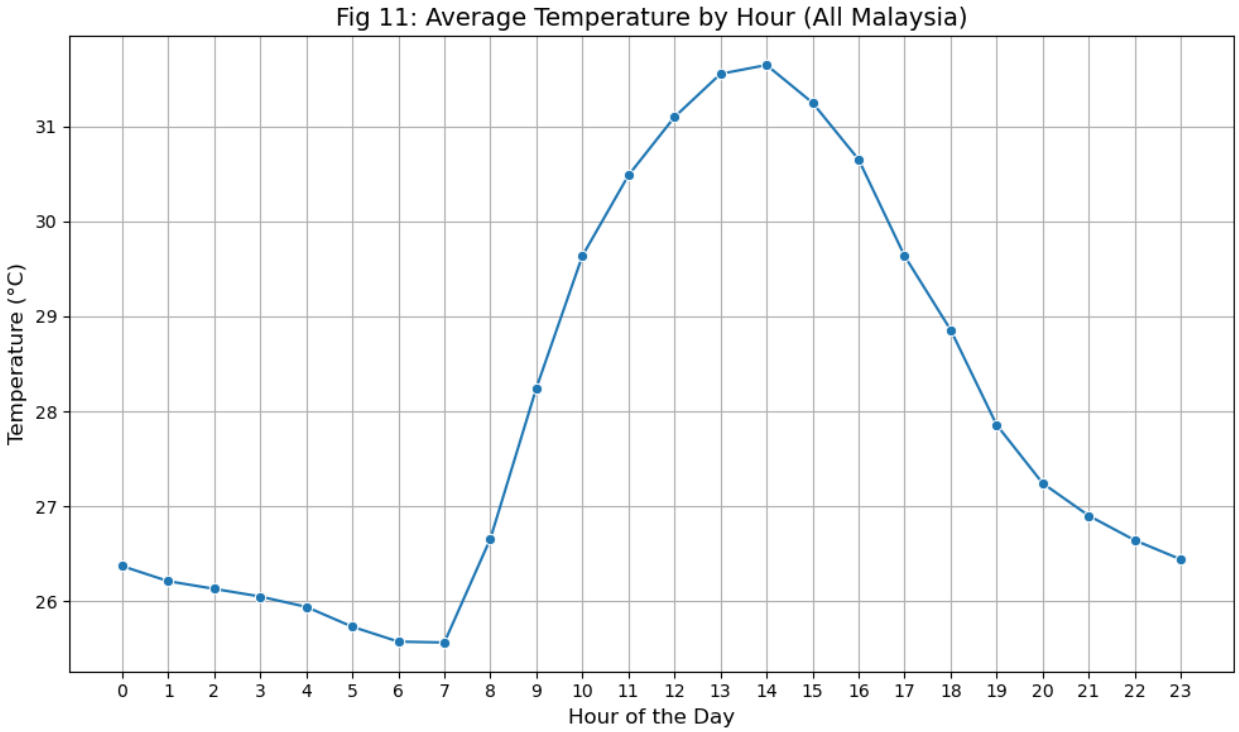


Fig 11: Average Temperature by Hour (All Malaysia)

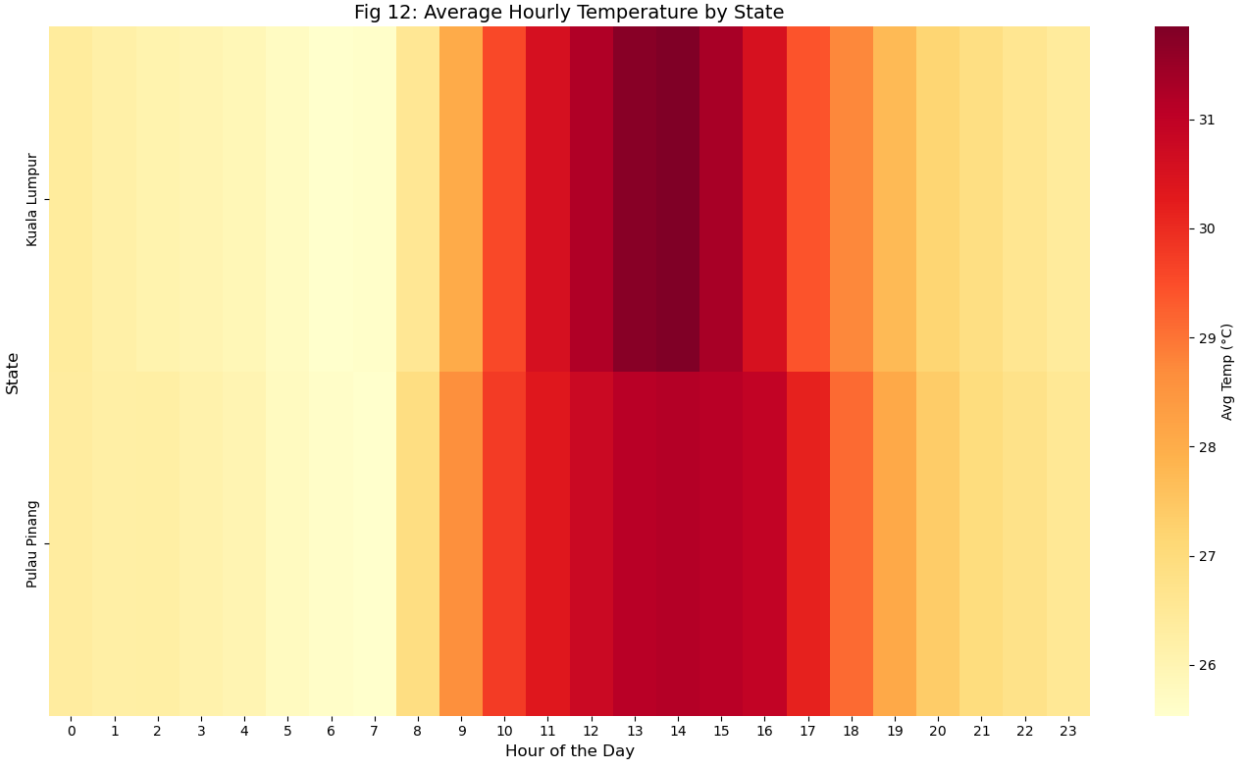


Fig 12: Average Hourly Temperature by State

Line Chart Analysis: Average Temperature by Hour

This line chart plots the average temperature across Malaysia for each hour of the day. The lowest temperature (~25.7°C) occurs around 6 to 7 a.m., just before sunrise.The temperature begins to rise steeply after 8 a.m., peaking around 2 to 3 p.m. at approximately 31.6°C.After 4 p.m., the temperature starts to decline gradually, returning to cooler levels (~26.5°C) by night (10 p.m. onward).The chart reflects a typical diurnal temperature cycle in tropical climates, with temperatures rising after sunrise and peaking in the early afternoon.The steep increase from 8 a.m. to 2 p.m. indicates strong solar radiation during that period.The sharp peak and symmetric decline suggest a consistent heat cycle with minimal anomaly across regions.

Heatmap Analysis: Average Hourly Temperature by State

This heatmap displays temperature across hours (x-axis) and states (y-axis), using color intensity to show average temperature (darker = hotter).Both Kuala Lumpur and Pulau Pinang exhibit similar color patterns: cooler temperatures from 12 a.m. to 7 a.m., heating up rapidly until 2–3 p.m., then cooling again. The most intense red appears around noon to 3 p.m., consistent with the peak seen in the line chart. The color intensity is fairly uniform between the two states, suggesting minimal regional temperature variation. The heatmap visually confirms the temporal trend from the line chart, with daytime heating and nighttime cooling. Interstate variation is minor, supporting the idea that Malaysia's temperature is more influenced by time of day than location. The data shows that despite being from different regions, both Kuala Lumpur and Pulau Pinang follow the same temperature rhythm.

Name: Ma Chenhuan

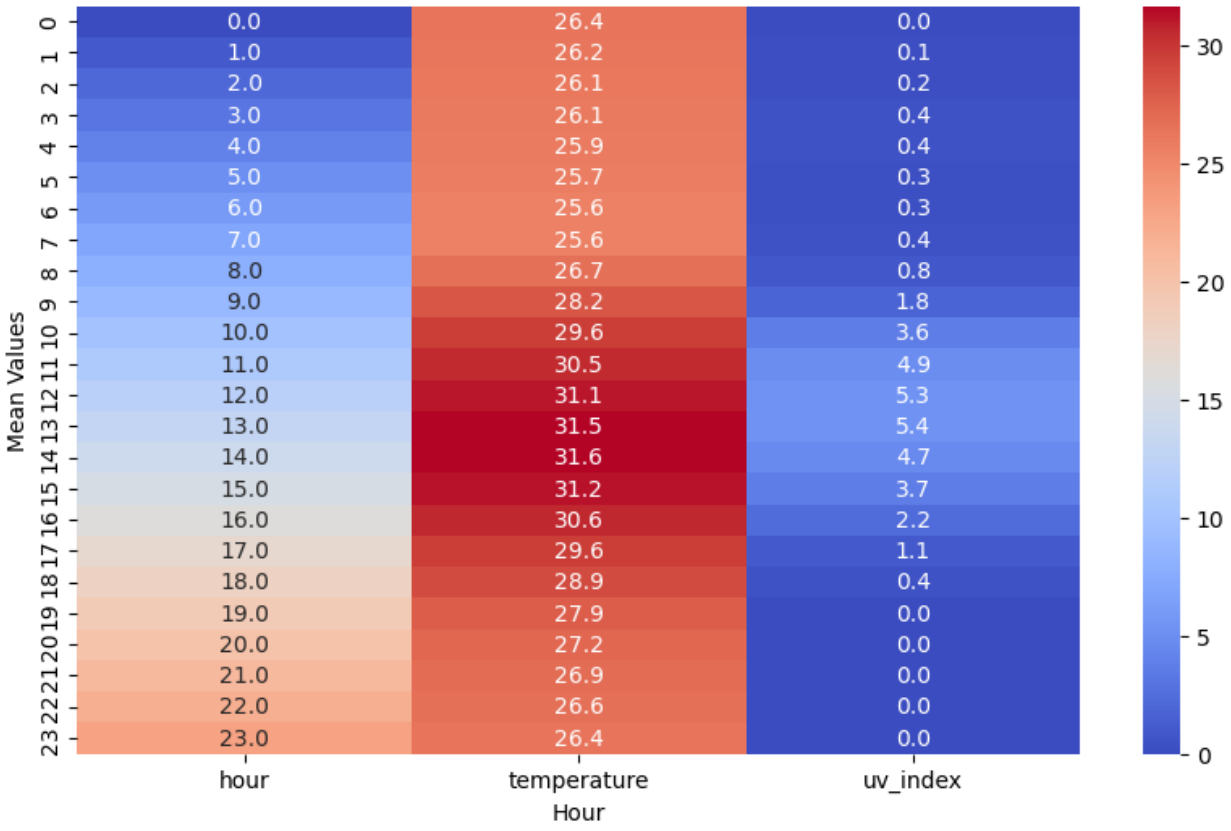
Question: Which time-of-day patterns emerge in temperature-UV index relationships?

The heatmap shows the daily temperature and UV index trends in various Malaysian cities. The x-axis shows hours, while the middle and right columns show temperature and UV index, respectively. Temperatures peak at noon (31.1-31.6°C) and drop in the early morning (25.6-26.4°C). The UV index follows the same pattern, peaking at noon and dropping in the morning and evening (near 0). The gradient from dark to light indicates values from low to high.

```
# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read the data
mw = pd.read_csv("malaysia_weather_data.csv")

# Hourly groupings and averages of temperature and UV index calculated
hourly_data = mw.groupby('hour').agg({'temperature': 'mean',
'uv_index': 'mean'}).reset_index()

# Create 24-hour cyclic heat maps
plt.figure(figsize=(10, 6))
sns.heatmap(hourly_data, annot=True, fmt=".1f", cmap='coolwarm')
plt.title('Fig 13: 24-hour circular heatmap (temp/UV correlation by
hour)')
```

```
plt.xlabel('Hour')
plt.ylabel('Mean Values')
plt.show()
```

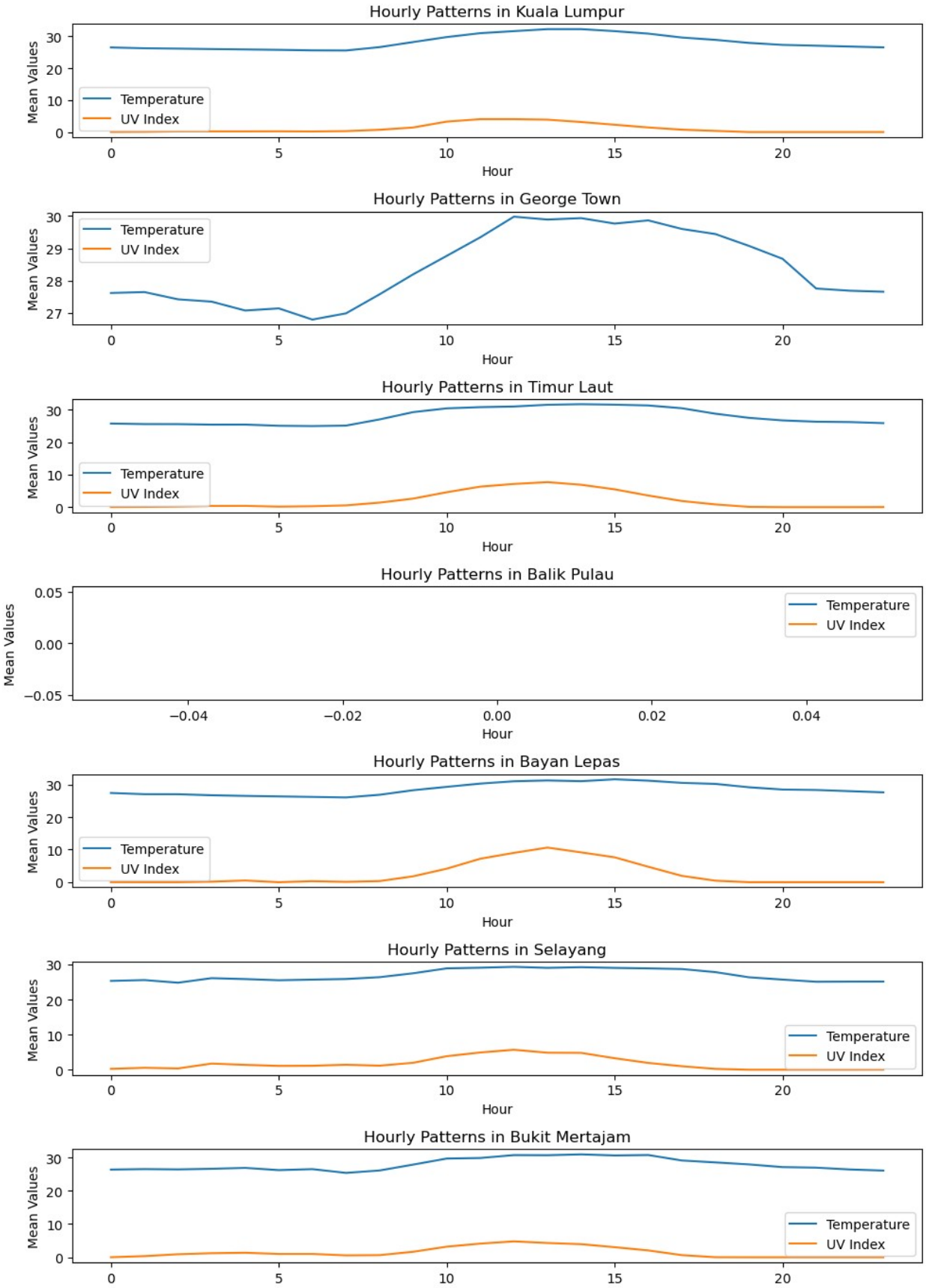
Fig 13: 24-hour circular heatmap (temp/UV correlation by hour)



In the second graph it can be seen that cities like Kuala Lumpur and Georgetown show afternoon temperature peaks with a slight UV rise at midday. Sandakan and Bali Batu Kau exhibit minimal fluctuations. Other cities follow a similar pattern with afternoon temperature and midday UV index increases.

```
# Get all cities
cities = mw['city'].unique()
nrows = len(cities)

# Create subgraph layouts
fig, axes = plt.subplots(nrows=nrows, ncols=1, figsize=(10, 2 *
nrows))
fig.subplots_adjust(hspace=0.5)

# Iterate over each city, plotting hourly temperatures and UV indices
for i, city in enumerate(cities):
    city_data = mw[mw['city'] == city]
    hourly_city_data = city_data.groupby('hour').agg({'temperature':
'mean', 'uv_index': 'mean'}).reset_index()
```

```
    # Plotting temperature line graphs
    sns.lineplot(data=hourly_city_data, x='hour', y='temperature',
ax=axes[i], label='Temperature')

    # Plotting UV index line graphs
    sns.lineplot(data=hourly_city_data, x='hour', y='uv_index',
ax=axes[i], label='UV Index')

    # Setting titles and labels
    axes[i].set_title(f'Hourly Patterns in {city}')
    axes[i].set_xlabel('Hour')
    axes[i].set_ylabel('Mean Values')
    axes[i].legend()

# Show charts
plt.tight_layout()
plt.show()
```


Hourly Patterns in Kuala Lumpur
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in George Town
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in Timur Laut
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in Balik Pulau
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in Bayan Lepas
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in Selayang
Mean Values
Hour
Temperature
UV Index
Hourly Patterns in Bukit Mertajam
Mean Values
Temperature
UV Index

Conclusion: The line chart effectively highlights Malaysia's daily temperature rhythm, peaking in mid-afternoon and cooling toward midnight. The heatmap supports this by showing that states follow a synchronized heating and cooling pattern. These charts confirm that time of day is the dominant factor in temperature variation across Malaysia, more so than location.

Name: Koay Ji Wei

Question: Can dew point accurately predict temperature fluctuations?

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
headers = [
    "Station", "District", "State", "Temperature", "Pressure",
"Dew_Point", "Humidity",
    "Rainfall", "Rainfall_2", "Max_Temperature", "Wind_Speed",
"Wind_Direction",
    "Wind_Gust", "Col1", "Year", "Month", "Day", "Hour", "Min", "Col2"
]
df = pd.read_csv("malaysia_weather_data.csv", names=headers,
header=None, dtype=str, low_memory=False)
df_2_percent = df.sample(frac=0.02, random_state=42)
# Convert Dew_Point and Temperature columns to numeric
df_2_percent['Dew_Point'] = pd.to_numeric(df_2_percent['Dew_Point'],
errors='coerce')
df_2_percent['Temperature'] =
pd.to_numeric(df_2_percent['Temperature'], errors='coerce')


# Convert relevant columns to numeric
for col in ['Temperature', 'Dew_Point', 'Humidity']:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Refining Data Set
df = df.dropna(subset=['Temperature', 'Dew_Point', 'Humidity',
'Month']).copy()

#Simple Linear Regression: Dew_Point → Temperature
x = df['Dew_Point'].values
y = df['Temperature'].values

# Using Sloper Intercept Form
slope, intercept = np.polyfit(x, y, 1)
df['Pred_Temp'] = slope * df['Dew_Point'] + intercept

# Make predictions
df['Pred_Temp'] = slope * df['Dew_Point'] + intercept
```

```
df['Residual'] = df['Temperature'] - df['Pred_Temp']

#Plot 1: Regression plot with confidence interval
plt.figure(figsize=(10, 6))
sns.regplot(x='Dew_Point', y='Temperature', data=df_2_percent,
scatter_kws={'alpha': 0.3}, line_kws={"color": "red"})
plt.title("Fig 14: Regression: Dew Point vs. Temperature")
plt.xlabel("Dew Point (°C)")
plt.ylabel("Temperature (°C)")
plt.tight_layout()
plt.show()

# Plot 2: Error by month
# Calculate Mean Absolute Error per month
df['Abs_Error'] = df['Residual'].abs()
monthly_error = df.groupby('Month')['Abs_Error'].mean().reset_index()

# Make month names into short form
month_map = {
    'January': 'Jan', 'February': 'Feb', 'March': 'Mar', 'April':
'Apr',
    'May': 'May', 'June': 'Jun', 'July': 'Jul', 'August': 'Aug',
    'September': 'Sep', 'Sept': 'Sep', 'October': 'Oct',
    'November': 'Nov', 'December': 'Dec',
    'JAN': 'Jan', 'FEB': 'Feb', 'MAR': 'Mar', 'APR': 'Apr', 'JUN':
'Jun', 'JUL': 'Jul',
    'AUG': 'Aug', 'SEP': 'Sep', 'OCT': 'Oct', 'NOV': 'Nov', 'DEC':
'Dec'
}
# Convert Month to numeric (e.g., '01' -> 1)
df['Month'] = pd.to_numeric(df['Month'], errors='coerce')
df = df[df['Month'].between(1, 12)]

# Calculate absolute prediction error
df['Abs_Error'] = (df['Temperature'] - df['Pred_Temp']).abs()

# Group by months numerically
monthly_error = df.groupby('Month')['Abs_Error'].mean().reset_index()

# Plot error per month
plt.figure(figsize=(10, 6))
sns.barplot(x='Month', y='Abs_Error', data=monthly_error,
color='skyblue')
plt.title('Fig 15: Prediction Error by Month (|Actual - Predicted|)')
plt.xlabel('Month')
plt.ylabel('Mean Absolute Error (°C)')
plt.xticks(ticks=np.arange(0, 12), labels=['Jan', 'Feb', 'Mar', 'Apr',
'May', 'Jun',
                                           'Jul', 'Aug', 'Sep', 'Oct',
'Nov', 'Dec'])
```

```
plt.tight_layout()
plt.show()
```

Fig 14: Regression: Dew Point vs. Temperature

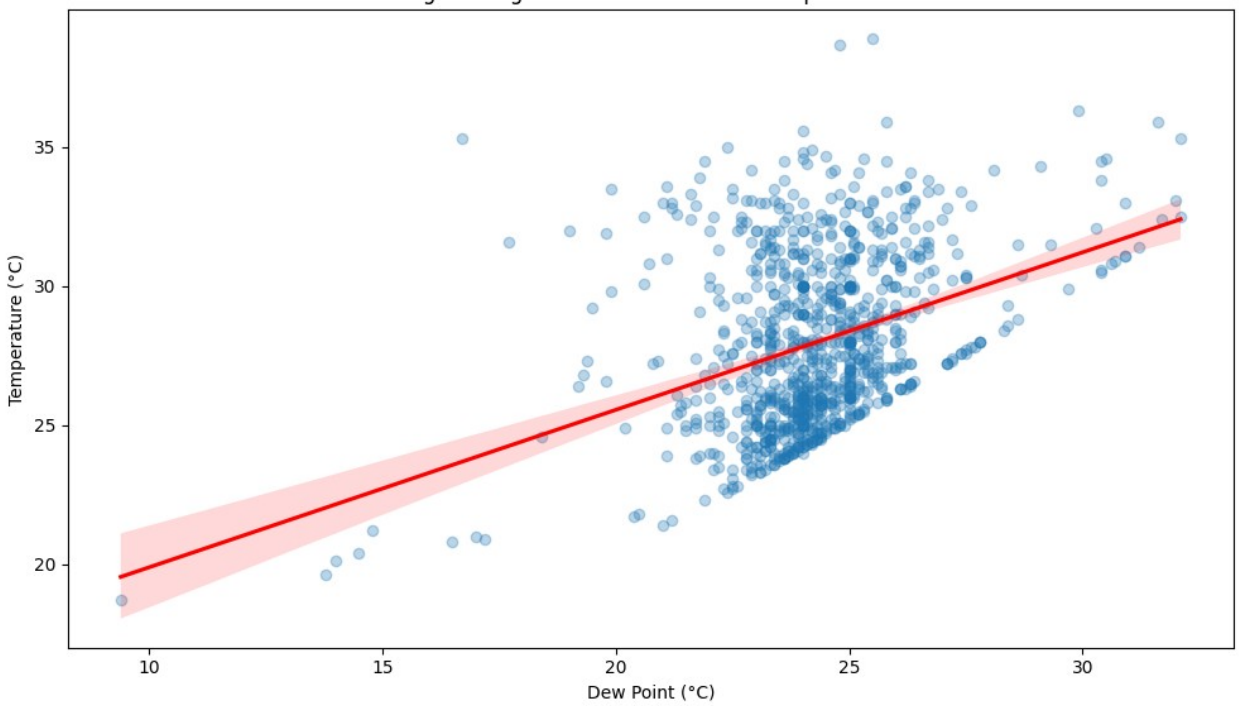


Fig 15: Prediction Error by Month (|Actual - Predicted|)

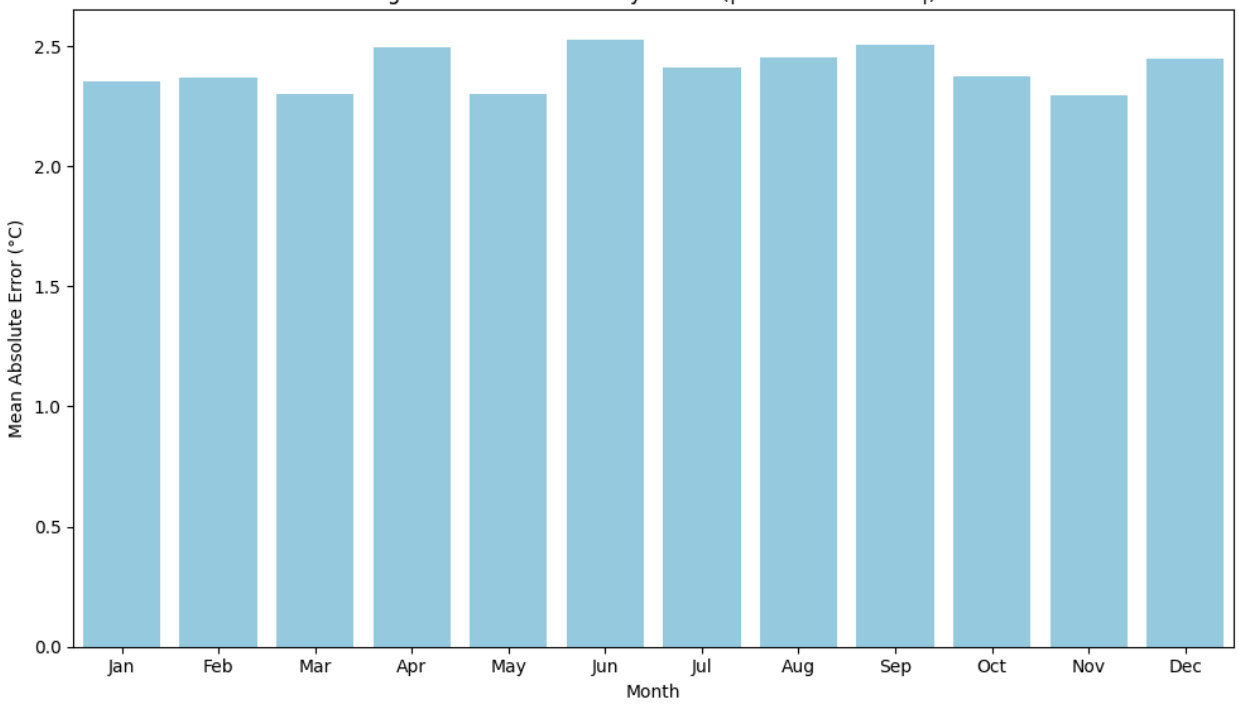


Fig 14:

Key Observations:

There is a clear positive linear relationship between drew point and temperature

The red regression line shows that as dew point increases, temperature also normally increases

Confidence interval that is shaded in red is relatively narrow around the central region which means it is more accurate to predict in that region

Data points are normally pretty concentrated except for some points that are outliers

Interpretation:

This graph supports that idea that dew points can be used to predict temperature, especially around the areas with higher confidence interval. However, there are outliers that causes the model to be less accurate in extreme cases.

Fig 15:

Key Observations:

The mean absolute error stays between around 2.3 and 2.6 celsius across the whole year

There are no drastic variation across all the months which shows that it is normally pretty consistent

Interpretation:

The model's prediction accuracy is fairly consistent throughout the whole year since there are no seasonal variation changes. Dew points remains reasonably stable predictor across all months which means it is relatively accurate to use dew points to predict.

Conclusion:

The analysis shows that dew point is a strong linear predictor of temperature, with consistent predictive power across the whole year. While it cannot produce exact prediction results, it is still a good foundation for temperature estimation for every month. Therefore, dew point can accurately predict temperature fluctuations to a reasonable extent with the exceptions of a couple outliers.

Report Conclusion

Report on Temperature-Related Factors in Malaysia

This report summarizes the analysis of temperature-related factors in Malaysia, covering aspects such as temperature, humidity, dew point, wind patterns, and UV index.

Temperature and Humidity

There is a negative correlation between temperature and humidity in Malaysia. The average temperatures across different regions are relatively similar, generally ranging between 20–30°C, while average humidity levels mostly fall within 60–80. Geographic location influences temperature, though the available data provides limited specific analysis. Current data does not clearly explain regional temperature variations based on wind patterns. Additionally, Penang exhibits relatively minor temperature fluctuations, which may be attributed to its geographical location and maritime moderation.

Wind Patterns and Temperature Relationship

The wind analysis revealed slight differences between regions, with coastal areas like Pulau Pinang experiencing higher average wind speeds and gust intensities compared to inland areas such as Kuala Lumpur. Although wind speed did not show a strong effect on temperature, the variation in gust intensity suggests that local geography, such as proximity to the coast, may influence wind behavior. Overall, wind patterns were relatively consistent across states and showed only a minor contribution to differences in local weather conditions.

Diurnal Temperature Patterns

Nationally, the average temperature typically peaks between time-frame 13:00 and 14:00. This pattern is consistent in states such as Kuala Lumpur and Penang, as clearly illustrated in hourly temperature charts.

Dew Point and Temperature Fluctuations

Regression analysis between dew point and temperature indicates a relationship, but monthly prediction errors remain relatively high, averaging around 2–2.5°C. This suggests that dew point may not precisely predict temperature fluctuations.

Temperature–UV Index Relationship

Regarding the diurnal variation pattern between temperature and UV index, in most regions such as Kuala Lumpur and George Town, both temperature and UV index begin to rise in the early morning, peak around noon or early afternoon, and then decline. A 24-hour heatmap further illustrates the correlation between temperature and UV index at different times.