Group 5

CLUSTERING ANALYSIS OF COVID-19 SPREAD IN MALAYSIA: IDENTIFYING HIGH-RISK REGIONS FOR CHILDREN AND ADOLESCENTS.

NO.	NAME	ID	COURSE
1	TAM KYLIE	21000451	COMPUTER SCIENCE
2	OOI CHIAO EE	21000422	COMPUTER SCIENCE
3	ANIS FARIDA BINTI AHMAD BAHARIN	22007538	COMPUTER SCIENCE
4	NUR AININ SOFIYA BINTI TUKIRAN	22008550	COMPUTER SCIENCE

Group 5

SUMMARY OF PROBLEMS AND SOLUTIONS.

Problem 1: Identifying High-Risk Regions for Children and Adolescents

- The goal is to pinpoint regions in Malaysia that pose a high risk for COVID-19 outbreaks among children and adolescents.
- This comprehensive evaluation will span across the diverse states of East Malaysia, North Malaysia, and West Malaysia.
- The identification of such high-risk areas is crucial for implementing targeted public health interventions to protect vulnerable populations.

Solution 1: K-Means Clustering

- Involves the implementation of a sophisticated clustering algorithm, notably the K-Means clustering methodology.
- By leveraging this algorithmic approach, the system aims to categorize states into three clusters: low risk, high risk, and moderate risk.
- This analytical process will group regions based on the similarity of patterns in the spread of COVID-19 among children and adolescents.

Group 5

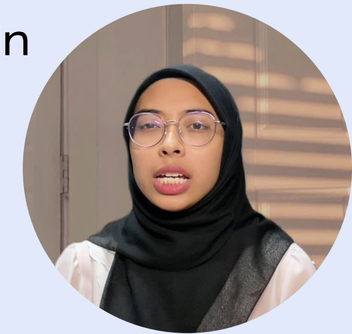
SUMMARY OF PROBLEMS AND SOLUTIONS.

Problem 2: Ensuring Stability and Robustness in Clustering Results

- An inherent challenge with K-Means clustering is its sensitivity to initial cluster center assignments, which can lead to variability in results.
- Ensuring the stability and robustness of the clustering solution is essential for reliable findings.

Solution 2: Iterative K-Means with Different Random Seeds and Visualize The Clusters for Comparison

- To address the stability concern, the K-Means algorithm is run iteratively 25 times.
- Each iteration uses a different random seed for the initialization of cluster centres.
- To facilitate a comprehensive understanding of the variability, the results of each iteration are visualized.
- Scatter plots are created to compare the clustering outcomes across different runs which allows for the identification of consistent patterns and aids in choosing a stable solution.



MOTIVATION AND BACKGROUND.

Background:

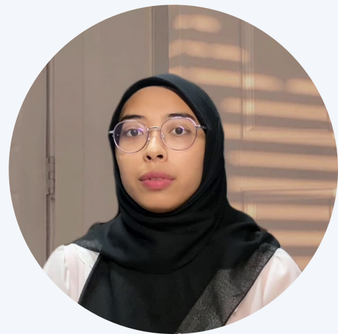
- Public health in the COVID-19 pandemic demands a focused approach, especially for vulnerable demographics like children and adolescents.
- Determining high-risk regions for COVID-19 outbreaks among children and adolescents.
- Paramount for targeted interventions, optimizing healthcare resources, and tailoring strategies to regional dynamics.

Significance of the Problem:

- Children and adolescents, though less susceptible, play a crucial role in community transmission.
- Unique interactions, behaviors, and potential long-term impacts make them distinct and requiring special attention.

Importance of Data Science:

- Clustering algorithms like k-means reveal patterns imperceptible through traditional analysis.



DATASET.

Source and Origin - sourced from the GitHub repository of the Malaysian Ministry of Health, available at https://github.com/MoH-Malaysia/covid19-public/blob/main/epidemic/cases_state.csv

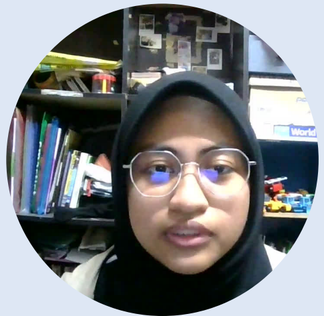
Data Format - CSV (Comma-Separated Values) format.

Variables and Columns - includes columns such as date, state, number of cases, deaths, number of cases recovered, number of cases for children, adolescent and elderly, and potentially other relevant information.

Temporal Coverage - spans from 25/01/2020 to 18/11/2023.

Geographical Coverage - covers all states in Malaysia and we group them into West Malaysia, East Malaysia and North Malaysia which has been stored in a variable called “Region”

Potential Uses - analysts can utilize this dataset to track the progression of COVID-19 cases, identify trends, and assess the impact on different states in Malaysia.



METHODOLOGY

i. Data preparation

- cases_state data is read as CSV file
- Sourced from the GitHub repository of the Malaysian Ministry of Health.

```
# Data preparation
# This is the covid-19 data from date 25/1/2020 to 18/11/2023
# Read as csv file which is a text file format that
# uses commas to separate values.
covid_data <- read.csv("C:/Users/60162/Downloads/cases_state.csv", header = TRUE)
```

METHODOLOGY

ii. Data exploration(descriptive)

- Understand the characteristics of the dataset which it helps to display the structure of dataset.

```
# Data Exploration(descriptive)
# To understand the characteristics of the dataset
# Display structure of the dataset
str(covid_data)
```

```
## 'data.frame':    22304 obs. of  26 variables:
##  $ date            : chr  "2020-01-25" "2020-01-25" "2020-01-25" "2020-01-25" ...
##  $ state           : chr  "Johor" "Kedah" "Kelantan" "Melaka" ...
##  $ cases_new       : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_import    : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_recovered : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_active    : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_cluster   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_unvax     : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_pvax      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_fvax      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_boost     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_child     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_adolescent: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_adult     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ cases_elderly   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_0_4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_5_11      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_12_17     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_18_29     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_30_39     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_40_49     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ cases_50_59     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_60_69     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_70_79     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_80        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region          : chr  "East Malaysia" "North Malaysia" "North Malaysia" "East Malaysia" ...
```

```
# Display the first few rows of the dataset
head(covid_data)
```

```
##         date            state cases_new cases_import cases_recovered
## 1 2020-01-25            Johor         4            4               0
## 2 2020-01-25            Kedah         0            0               0
## 3 2020-01-25         Kelantan         0            0               0
## 4 2020-01-25           Melaka         0            0               0
## 5 2020-01-25 Negeri Sembilan         0            0               0
## 6 2020-01-25           Pahang         0            0               0
##   cases_active cases_cluster cases_unvax cases_pvax cases_fvax cases_boost
## 1            4             0           4          0          0           0
```

METHODOLOGY

ii. Data exploration(descriptive)

```
# Display the class of the dataset
class(covid_data)
```

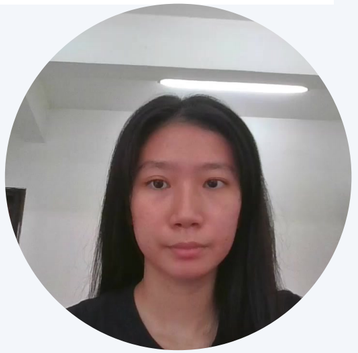
```
## [1] "data.frame"
```

```
# Display summary statistics of the dataset
summary(covid_data)
```

```
##      date               state              cases_new        cases_import
##  Length:22304       Length:22304       Min.   :    0.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.:    2.0   1st Qu.:  0.000
##  Mode  :character   Mode  :character   Median :   26.0   Median :  0.000
##                                        Mean   :  230.3   Mean   :  1.747
##                                        3rd Qu.:  171.0   3rd Qu.:  0.000
##                                        Max.   :11692.0   Max.   :351.000
##  cases_recovered    cases_active     cases_cluster      cases_unvax
##  Min.   :    0.0   Min.   :  -630   Min.   :   0.00   Min.   : -1.0
##  1st Qu.:    2.0   1st Qu.:    54   1st Qu.:   0.00   1st Qu.:  0.0
##  Median :   24.0   Median :   500   Median :   0.00   Median :  6.0
##  Mean   :  228.2   Mean   :  2790   Mean   :  23.85   Mean   : 90.7
##  3rd Qu.:  162.0   3rd Qu.:  2315   3rd Qu.:   9.00   3rd Qu.: 50.0
##  Max.   :12379.0   Max.   :103574   Max.   :1545.00   Max.   :6112.0
##    cases_pvax        cases_fvax       cases_boost       cases_child
##  Min.   :   0.00   Min.   :   0.00   Min.   :   0.00   Min.   :   0.00
##  1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00
##  Median :   0.00   Median :   2.00   Median :   0.00   Median :   2.00
```

```
# Display the class of each variable in the dataset
sapply(covid_data, class)
```

```
##             date            state        cases_new     cases_import
##      "character"      "character"        "integer"        "integer"
##  cases_recovered     cases_active     cases_cluster       cases_unvax
##        "integer"        "integer"        "integer"        "integer"
##       cases_pvax       cases_fvax      cases_boost      cases_child
##        "integer"        "integer"        "integer"        "integer"
## cases_adolescent      cases_adult     cases_elderly        cases_0_4
##        "integer"        "integer"        "integer"        "integer"
##       cases_5_11      cases_12_17      cases_18_29      cases_30_39
##        "integer"        "integer"        "integer"        "integer"
##      cases_40_49      cases_50_59      cases_60_69      cases_70_79
##        "integer"        "integer"        "integer"        "integer"
##         cases_80           Region
##        "integer"      "character"
```

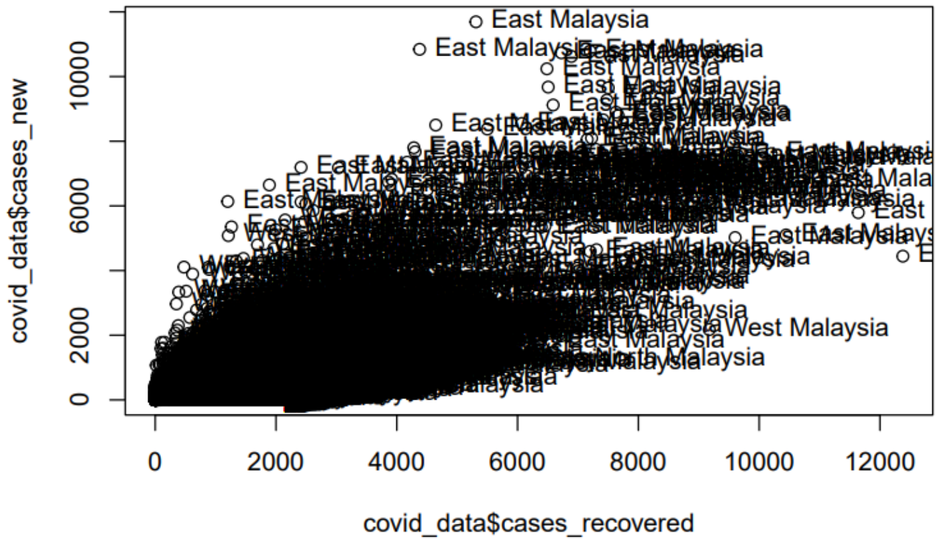


METHODOLOGY

ii. Data exploration(descriptive)

- Then, take a look into the visualisation of the new cases and recovered cases through scatter plot to know the relationship between these two cases.

```
# Take a look at the scatter plot for new cases(cases_new)
# and recovered cases(cases_recovered)
plot(covid_data$cases_new ~ covid_data$cases_recovered, data = covid_data)
with(covid_data, text(covid_data$cases_new~ covid_data$cases_recovered,
                      labels = covid_data$Region, pos = 4))
```



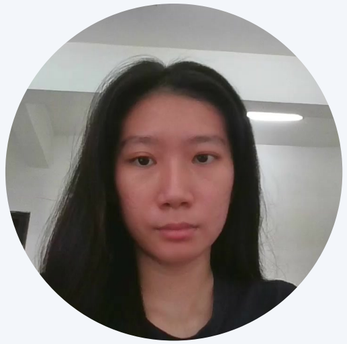
METHODOLOGY

iiii. Data preprocessing

- This involves categorizing states into ‘Regions’ variable (East Malaysia, North Malaysia, West Malaysia).
- Remove the Nas and missing values to ensure data quality.

```r
# Data preprocessing
# Create a 'Region' Variable based on the 'state' variable
# Classify regions based on the states
covid_data <- covid_data %>%
  mutate(Region = case_when(
    state %in% c("Johor", "Melaka", "Negeri Sembilan",
                 "Pahang", "Selangor", "Terengganu",
                 "W.P. Kuala Lumpur", "W.P. Labuan", "W.P. Putrajaya") ~ "East Malaysia",
    state %in% c("Kedah", "Perlis", "Pulau Pinang", "Kelantan", "Perak") ~ "North Malaysia",
    state %in% c("Sabah", "Sarawak") ~ "West Malaysia",
    TRUE ~ "Other"
  ))
```

```r
# Remove rows with NAs
# Remove rows with missing values to ensure data quality
covid_data <- na.omit(covid_data)
```



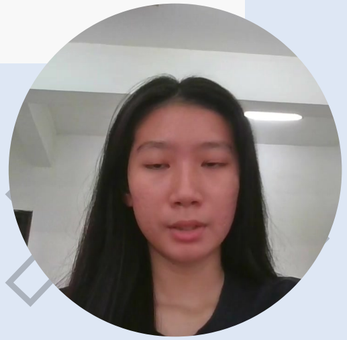
METHODOLOGY

iiii. Data preprocessing

- Select the features needed for clustering analysis. I
- We select state, Region, cases_child and cases_adolescent

```r
# Select the features for analysis
# Select relevant features for clustering analysis
selected_features <- covid_data %>%
  filter(Region %in% c("East Malaysia", "North Malaysia", "West Malaysia")) %>%
  select(state, Region, cases_child, cases_adolescent)

# Drop rows with missing values
# Remove any remaining rows with missing values in the selected features
selected_features <- selected_features %>%
  drop_na()
```



METHODOLOGY

iv. Data Normalization

- Normalize the selected features by using scale() and standardize them for better performance in k-means clustering.
- Calculate distance matrix between observations in the normalized features.

```r
# Normalize the data
# Standardize the selected features for better performance in k-means clustering
normalized_features <- scale(selected_features[, c("cases_child", "cases_adolescent")])

# Calculate distance matrix
# Calculate the pairwise distances between observations in the normalized features
distance = dist(normalized_features)
```

METHODOLOGY

v. Elbow Method

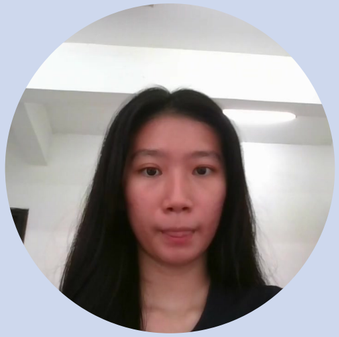
- It helps to determine the optimal number of clusters using the elbow method.
- We create another features which contains only numeric value and iterate through different cluster numbers.
- Plot Elbow Method and visualize the within-cluster sum of squares for different cluster numbers

```r
# Elbow Method
# Determine the optimal number of clusters using the elbow method
wcss_values <- numeric(10)

#create another features which contains only numeric value
wanted_features <- covid_data[,c("cases_child", "cases_adolescent")]

# Iterate through different cluster numbers
for (i in 1:10) {
  kmeans_model <- kmeans(wanted_features, centers = i)
  wcss_values[i] <- kmeans_model$tot.withinss
}

# Plot Elbow Method
# Visualize the within-cluster sum of squares for different cluster numbers
plot(1:10, wcss_values, type = "b", pch = 19, frame = FALSE, main = "Elbow Method",
     xlab = "Number of Clusters", ylab = "Within-Cluster Sum of Squares")
```



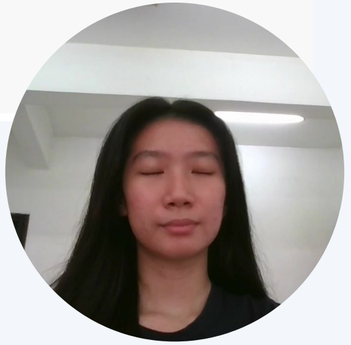
METHODOLOGY

vi. K-Means Clustering

- Determine the number of clusters and perform the k-means clustering.
- set.seed(200) is for reproducibility.
- We also performs k-means clustering on the dataset normalized_features with 3 clusters.
- Then we combine the selected features with the cluster assignments from the k-means model.

```r
# Determine the number of clusters and perform the k-means clustering
set.seed(200)
k <- 3
kmeans_model <- kmeans(normalized_features, centers = k)

# Add cluster assignments to the original dataset
segmented_data <- cbind(selected_features, Cluster = kmeans_model$cluster)
```



METHODOLOGY

vi. K-Means Clustering

- K-Means clustering is applied to identify three clusters (Low Risk, Moderate Risk, High Risk) based on COVID-19 cases in children and adolescents.
- Visualize Clusters by creating a scatter plot to visualize the clustering results, faceted by 'Region'.

```
# Create a new variable 'cluster_names' for descriptive cluster names
cluster_names <- c("Low Risk", "Moderate Risk", "High Risk")

# Add descriptive cluster names
segmented_data <- mutate(segmented_data, ClusterName = cluster_names[Cluster])



# Visualize Clusters
# Create a scatter plot to visualize the clustering results, faceted by 'Region'
ggplot(segmented_data, aes(x = cases_child, y = cases_adolescent, col = ClusterName)) +
  geom_point() +
  facet_wrap(~Region) +
  labs(title = "K-Means Clustering of COVID-19 Cases by Region",
       x = "Cases in Child Age Group", y = "Cases in Adolescent Age Group")
```

METHODOLOGY

vii. Iteration for 25 times

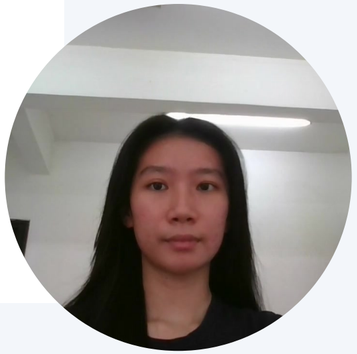
- The clustering process is repeated 25 times with different random seeds for stability assessment.
- Applies the k-means clustering process iteratively for different random starts, storing the cluster assignments for each iteration in the matrix wss.

```r
# Iterative K-Means and Visualization
# Perform iterative k-means with different random starts and visualize results
n_iterations <- 25
wss <- sapply(1:n_iterations, function(iteration){
  #Set a different random seed for each iteration
  set.seed(iteration)
  kmeans_model <- kmeans(wanted_features, centers = k, nstart = 1)  # nstart = 1 for reproducibility

  # Access cluster assignments
  cluster_assignments <- kmeans_model$cluster

  # Return the cluster assignments for this iteration
  return(cluster_assignments)
})

#Display the first few rows of wss
head(wss)
```



METHODOLOGY

vii. Iteration for 25 times

- Set up a 2x2 grid for subplots.
- Iterates through each iteration, creating scatter plots with consistent colors for clusters. Each plot represents the clustering result for a specific iteration.

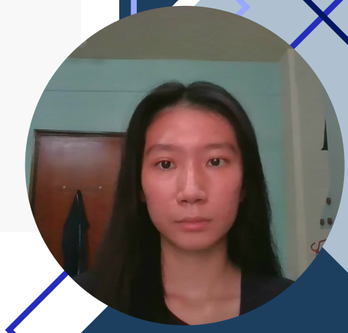
```
# Define a color palette for clusters
cluster_colors <- c("red", "green", "blue")

# Visualize the results (example for 2D data)
par(mfrow=c(2, 2))  # Set up a 2x2 grid for subplots

for (i in 1:n_iterations) {

  # Scatter plot with consistent colors for clusters
  plot(wanted_features, col = cluster_colors[wss[, i]], main = paste("Iteration", i),
       xlab = "cases_child", ylab = "cases_adolescent")

}
```



METHODOLOGY

viii. Display the cluster analysis result

- Check the size of each clusters
- Identify and display the high-risk regions and their counts in each clusters

```
# Print out the cluster analysis and result
# Check the size of each cluster
table(segmented_data$ClusterName)
```

```
##
##     High Risk      Low Risk Moderate Risk
##          1574         20550           180
```

```
# Identify the high-risk regions and their counts in each cluster
high_risk_regions <- segmented_data %>%
  filter(ClusterName == "High Risk") %>%
  group_by(Region) %>%
  tally()

# Display the high-risk regions and their counts
print("High-Risk Regions and Their Counts:")
```

```
## [1] "High-Risk Regions and Their Counts:"
```

```
print(high_risk_regions)
```

```
## # A tibble: 3 x 2
##   Region              n
##   <chr>           <int>
## 1 East Malaysia     855
## 2 North Malaysia    518
## 3 West Malaysia     201
```

METHODOLOGY

viii. Display the cluster analysis result

- Identify and display the region with the highest risk in the High Risk cluster

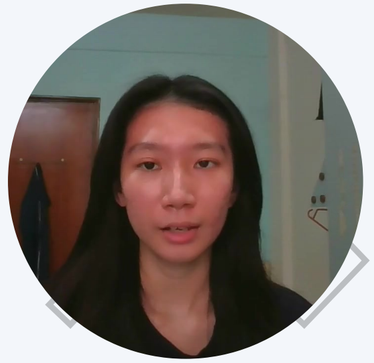
```
# Identify the region with the highest risk in the High Risk cluster
max_risk_region <- segmented_data %>%
  filter(ClusterName == "High Risk") %>%
  group_by(Region) %>%
  tally() %>%
  arrange(desc(n)) %>%
  slice(1)

# Display the region with the highest risk in the High Risk cluster
print("Region with the Highest Risk in High Risk Cluster:")
```

```
## [1] "Region with the Highest Risk in High Risk Cluster:"
```

```
print(max_risk_region)
```

```
## # A tibble: 1 x 2
##   Region            n
##   <chr>         <int>
## 1 East Malaysia   855
```

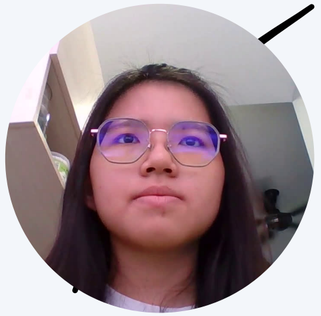


RESULT

```
str(covid_data)
```

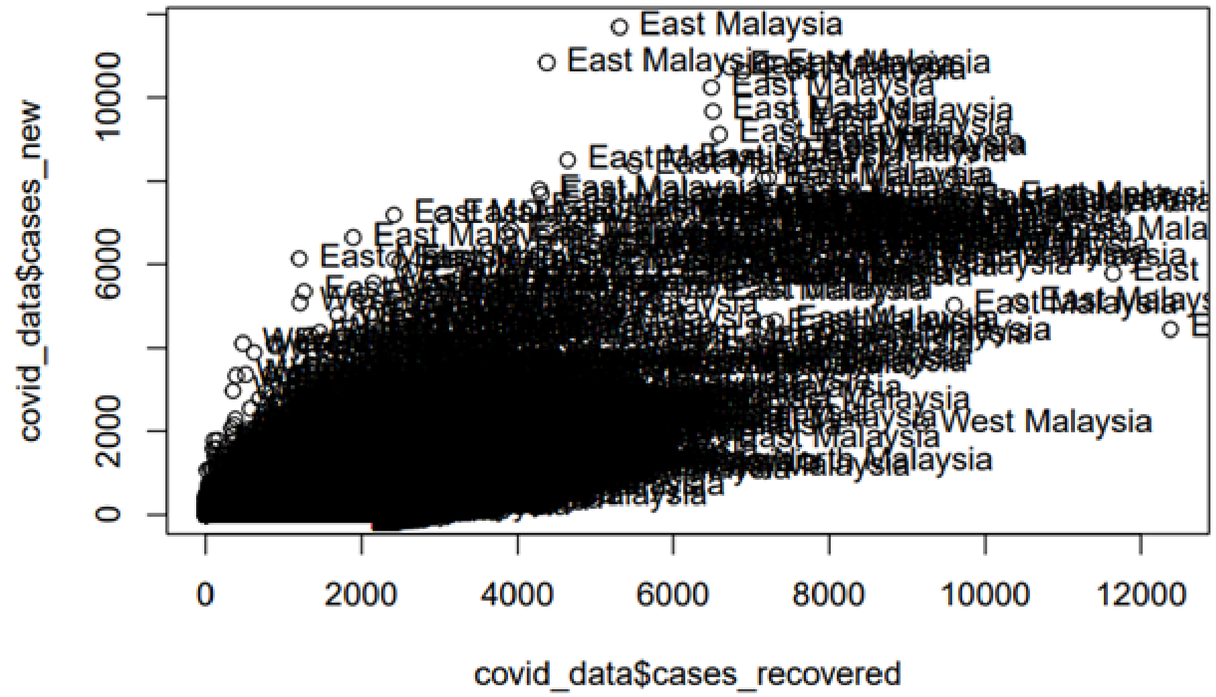
```
## 'data.frame':    22304 obs. of  26 variables:
##  $ date            : chr  "2020-01-25" "2020-01-25" "2020-01-25" "2020-01-25" ...
##  $ state           : chr  "Johor" "Kedah" "Kelantan" "Melaka" ...
##  $ cases_new       : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_import    : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_recovered : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_active    : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_cluster   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_unvax     : int  4 0 0 0 0 0 0 0 0 0 ...
##  $ cases_pvax      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_fvax      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_boost     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_child     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_adolescent: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_adult     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ cases_elderly   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_0_4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_5_11      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_12_17     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_18_29     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_30_39     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_40_49     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ cases_50_59     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_60_69     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_70_79     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cases_80        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region          : chr  "East Malaysia" "North Malaysia" "North Malaysia" "East Malaysia" ...
```

For the data exploration using str(), we can see the 'state' is grouped into 3 different regions and stored in the Region variables.



RESULT

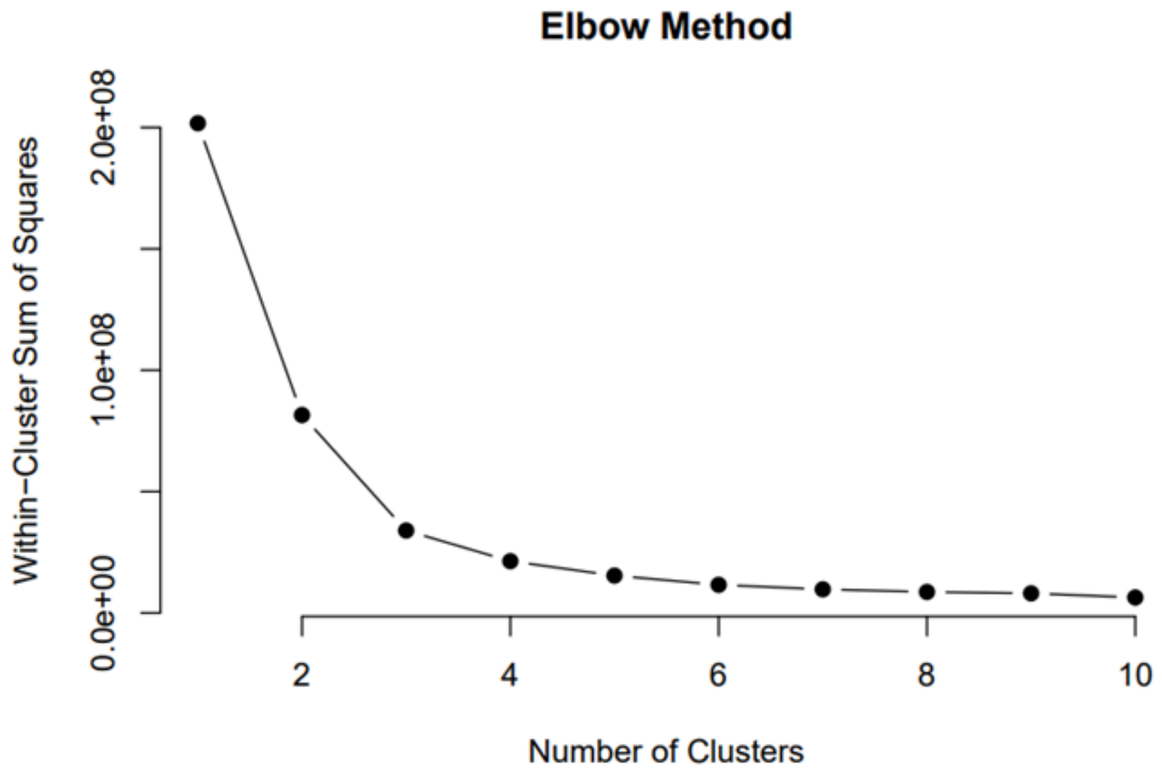
```
# Take a look at the scatter plot for new cases(cases_new)
# and recovered cases(cases_recovered)
plot(covid_data$cases_new ~ covid_data$cases_recovered, data = covid_data)
with(covid_data, text(covid_data$cases_new~ covid_data$cases_recovered,
                      labels = covid_data$Region, pos = 4))
```



- Scatter plot for new cases and recovered cases based on region which represents the relationship between new COVID-19 cases and recovered COVID-19 cases.
- Allow us to visualize this data by placing one variable (new cases) on the x-axis and another variable (recovered cases) on the y-axis.
- Each dot represents a region with a unique set of values for new cases and recovered cases.

RESULT

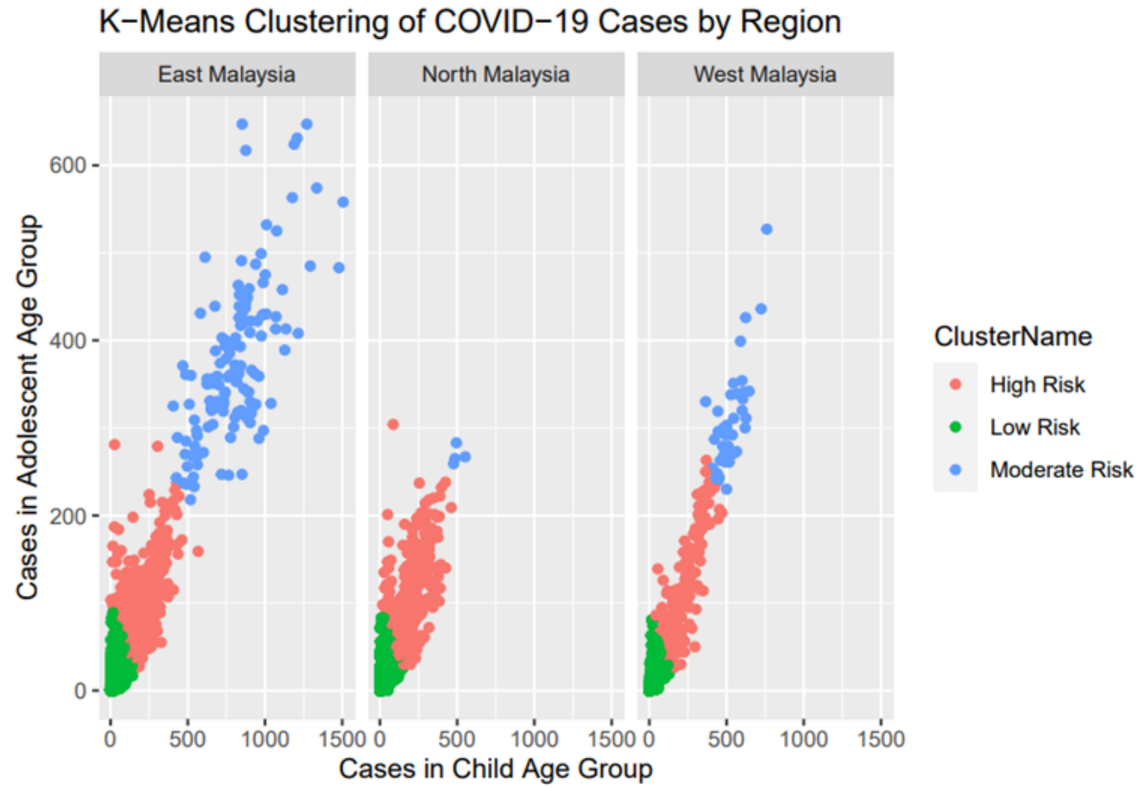
```r
# Plot Elbow Method
# Visualize the within-cluster sum of squares for different cluster numbers
plot(1:10, wcss_values, type = "b", pch = 19, frame = FALSE, main = "Elbow Method",
     xlab = "Number of Clusters", ylab = "Within-Cluster Sum of Squares")
```



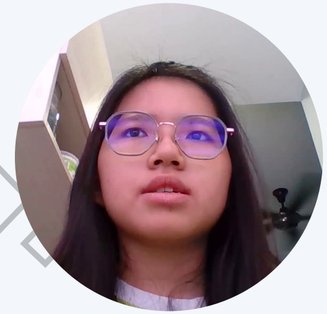
- Elbow Method is a widely used approach in clustering to determine the optimal number of clusters.
- visualizes the within-cluster sum of squares (WCSS) for different cluster numbers.
- It suggests the optimal number of clusters when the WCSS value starts to decrease at a slower rate.
- It indicates that the optimal number of clusters is 3, as the WCSS value starts to decrease at a slower rate for this cluster number.
- Result: The data can be separated into 3 distinct subgroups, which may provide a higher level of abstraction and interpretation.

RESULT

```
# Visualize Clusters
# Create a scatter plot to visualize the clustering results, faceted by 'Region'
ggplot(segmented_data, aes(x = cases_child, y = cases_adolescent, col = ClusterName)) +
  geom_point() +
  facet_wrap(~Region) +
  labs(title = "K-Means Clustering of COVID-19 Cases by Region",
       x = "Cases in Child Age Group", y = "Cases in Adolescent Age Group")
```

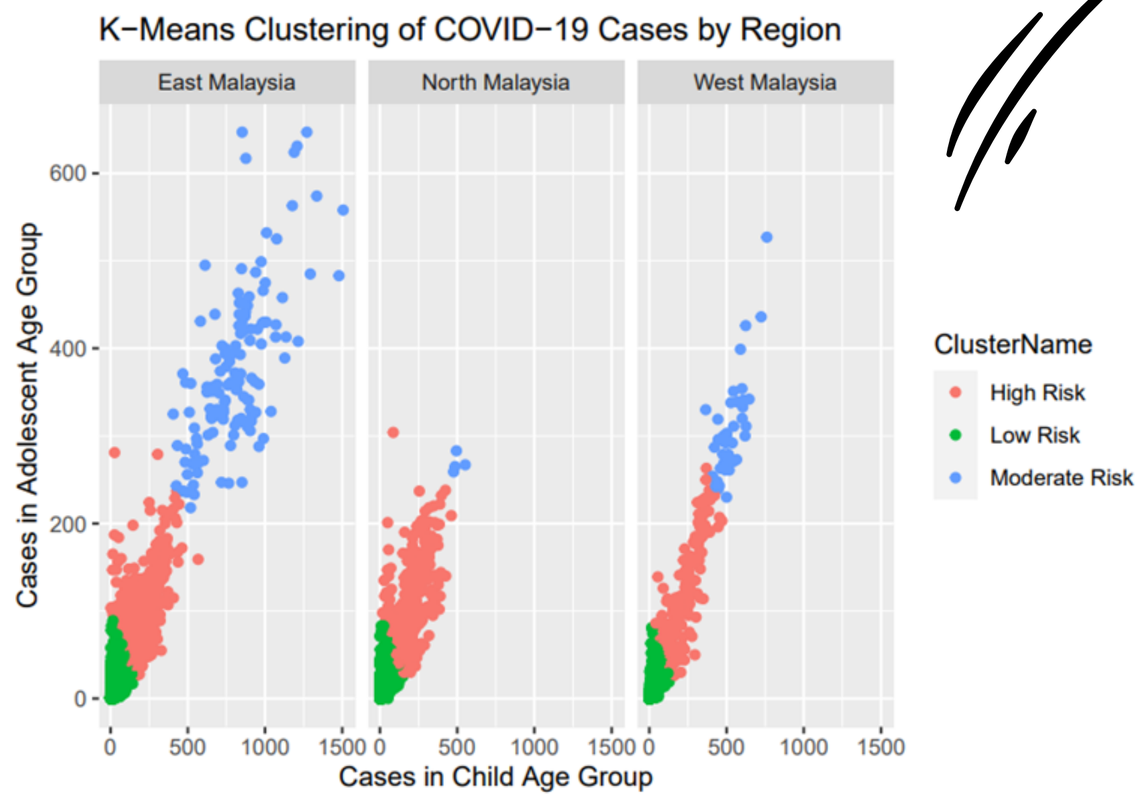


- The X-axis represents the number of COVID-19 cases in the child age group.
- The Y-axis represents the number of cases in the adolescent age group.
- Each dot represents a segment of data from the segmented_data dataset.
- The dots are coloured according to their assigned cluster.
- By examining the density and distribution of dots in each region, we can get a sense of how COVID-19 cases are clustered in that region.



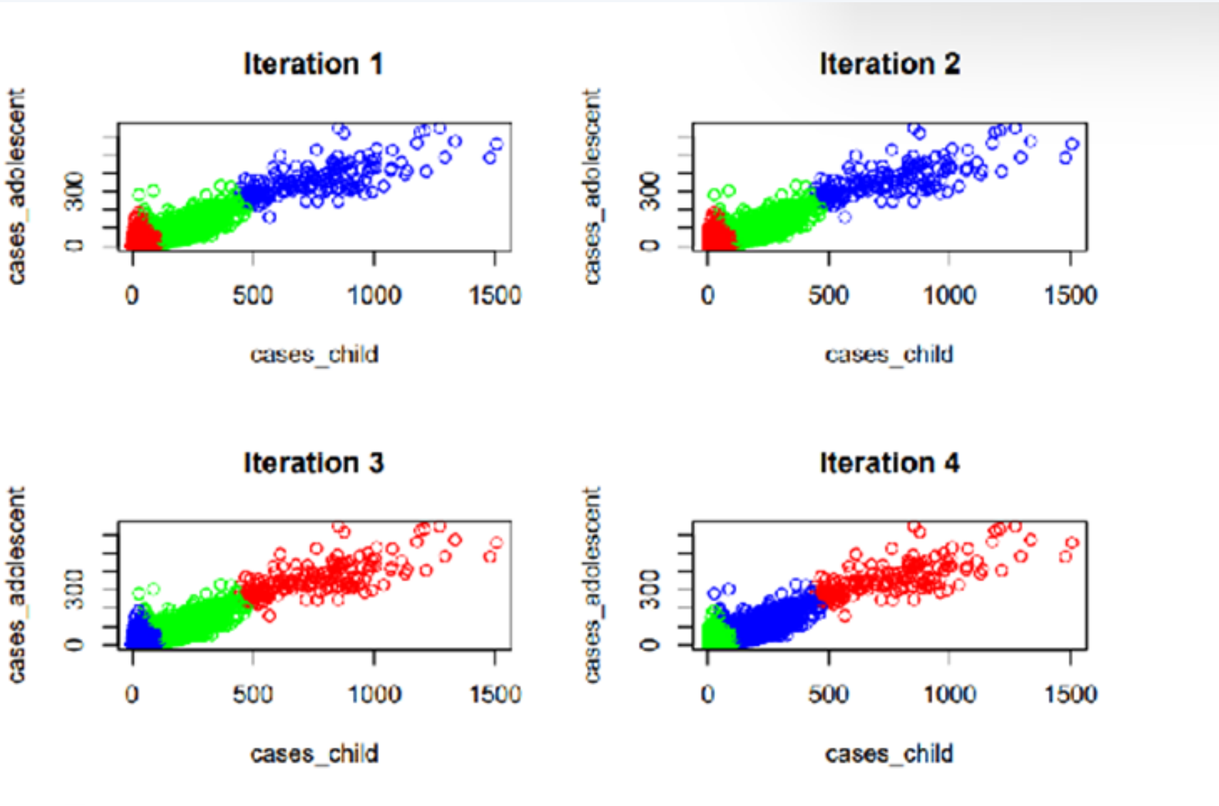
RESULT

```r
# Visualize Clusters
# Create a scatter plot to visualize the clustering results, faceted by 'Region'
ggplot(segmented_data, aes(x = cases_child, y = cases_adolescent, col = ClusterName)) +
  geom_point() +
  facet_wrap(~Region) +
  labs(title = "K-Means Clustering of COVID-19 Cases by Region",
       x = "Cases in Child Age Group", y = "Cases in Adolescent Age Group")
```

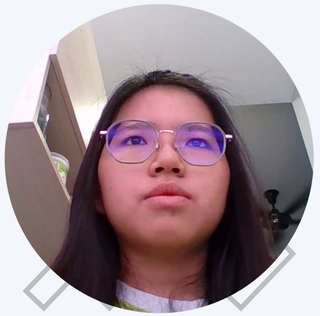


- This classification is based on the results of the K-Means clustering algorithm applied to the data.
- The algorithm is used to partition the data into a predefined number of clusters, with each cluster being represented by a centroid.
- In this case, the number of clusters is determined by the 'ClusterName' variable.
- For example, in the 'East Malaysia' facet, the cluster labeled 'High Risk' appears to have the highest number of cases in the child age group.
- This could suggest that this region is particularly vulnerable to COVID-19 due to its high-risk population.

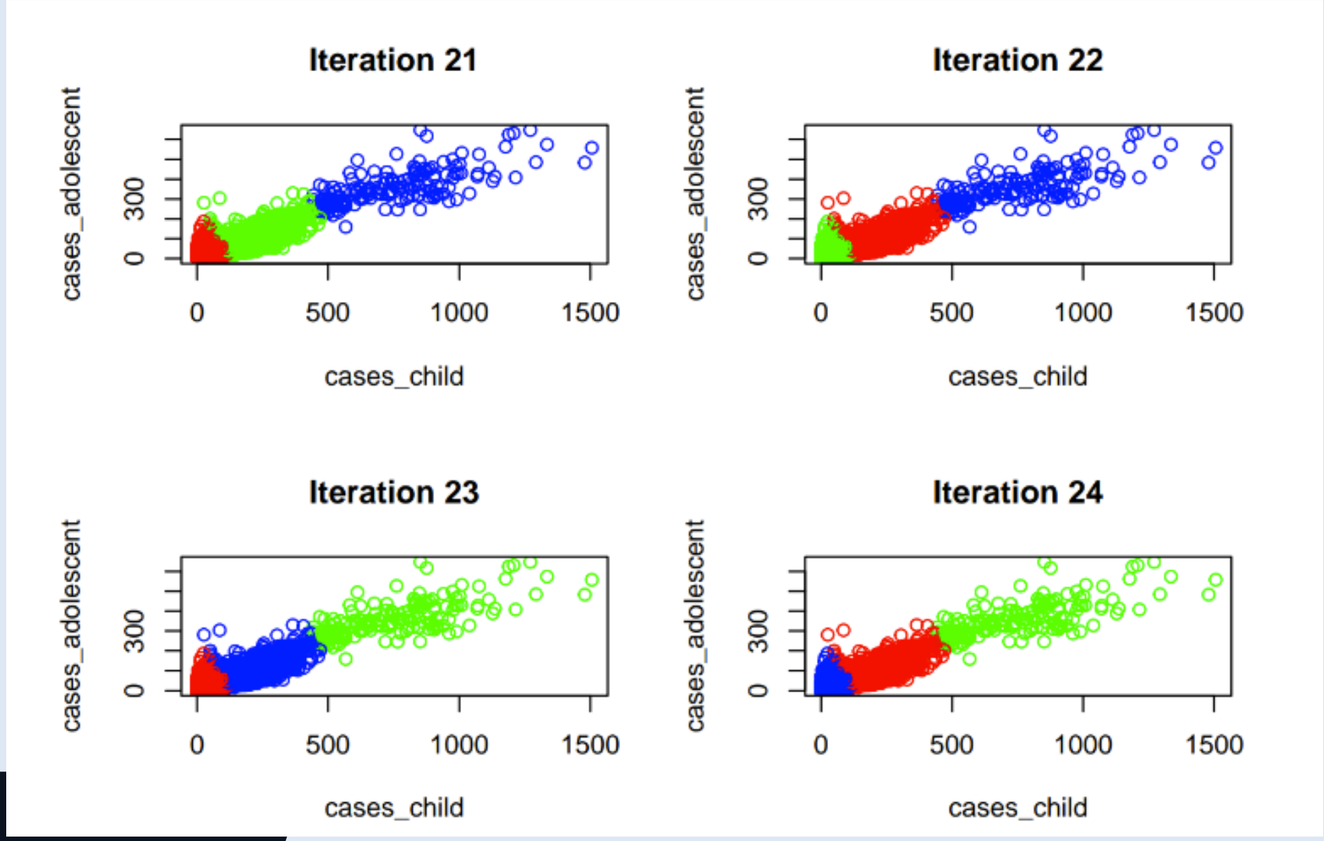
RESULT



- It will have 25 iterations.
- Pattern remains the same across 25 iterations in a K-Means clustering analysis
- Indicates that the algorithm has converged to a stable solution, and the clusters are consistent.
- This stability is generally a positive outcome, as it suggests that the identified clusters are robust and not highly sensitive to random initialization.
- K-Means is an iterative algorithm that aims to minimize the within-cluster sum of squares. If the algorithm converges to a stable solution early in the iterations, additional iterations may not significantly alter the clustering.



RESULT



- The clustering analysis reveals distinct patterns of COVID-19 spread in different regions.
- The results indicate specific regions categorized as High Risk for children and adolescents, providing valuable information for targeted public health interventions.
- The iterative approach enhances the robustness of the findings.

Group 5

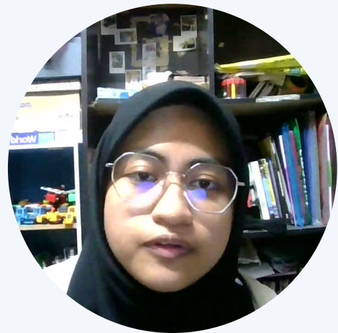
INDIVIDUAL REFLECTION

What did you learn from this project?

- This project on the clustering analysis of COVID-19 spread in Malaysia has been a rich learning experience, providing insights into both the technical aspects of data science and the application of these skills in the critical domain of public health.
- We significantly enhanced the proficiency in data preprocessing, feature engineering, and the practical implementation of clustering algorithms, particularly K-Means.
- Sensitivity of clustering algorithms to initializations became evident during the iterative analysis, underscoring the importance of algorithm stability.
- Working with a real dataset from the Malaysian Ministry of Health exposed us to the challenges of handling diverse and extensive datasets, refining our data exploration and visualization skills.

What do you wish you had known before you started?

- A deeper understanding of the socio-economic and cultural factors influencing COVID-19 spread in Malaysia would have enriched the interpretation of clustering results.
- Exploring alternative clustering techniques beyond K-Means might have provided additional perspectives on the data, a lesson that underscores the importance of diversifying analytical methodologies.



Group 5

INDIVIDUAL REFLECTION

What would you do differently?

- Allocate more time to initial dataset exploration, fostering a comprehensive understanding of its intricacies.
- Experiment with different cluster numbers in K-Means, iteratively refining the model to achieve a more nuanced interpretation of regional risk levels.

What advice would you offer to future students embarking on this project?

- Striking a balance between technical skills and domain knowledge, particularly in understanding the specific nuances of the dataset.
- Exploring alternative clustering techniques and maintaining thorough documentation of each analysis step are crucial for achieving more robust and informed results.
- Embracing a mindset of continuous learning and staying informed about new methodologies ensures adaptability in the rapidly evolving field of data science.
- This project has not only strengthened our technical capabilities but also deepened our appreciation for the impactful role of data science in addressing real-world challenges, especially in the realm of public health.

Group 5