Marba Iquen I **BSIT - 3C**
Prof:Edward B. Bertulfo **IT 353**

HANDS-ON ACTIVITY # 8

Let's up it up another level and use what you have learned from the entire course and try to predict future events just be processing the dataset and help in providing relevant information that will help with the decision making process.

Being *Mang Tani*

Scenario:

Your team is organizing an outdoor charity event in the coming months. To ensure the event is successful, you need to predict the likelihood of rain based on historical weather patterns. You have received a dataset of weather conditions over the past three months from a local weather station.

Objectives:

- Perform Data Exploration [*EDA*] on the provided dataset.
- Demonstrate how to execute Data Cleaning and Data Wrangling.
- Create Data Visualization.
- Produce Prediction(s).
- Provide Recommendation(s).

Preparing the Dataset and Libraries:

Before you can do anything, let's create a mock dataset using on of the libraries in the *tidyverse* packages, **tibble**. This library works similarly to the **data.frame()** function but, just like with rest of the *tidyverse* package, it is aimed to better simplify the workflow of the data scientist by allowing it to be more consistent and much easier to use.

```
# Install and Load the tidyverse library
install.packages(‘tidyverse’) # run the install.package() only if the library is not
installed yet  library(tidyverse) # this will load the complete tidyverse package
(dplyr, tidyr, ggplot2, tibble)  # Create mock weather dataset
set.seed(123)

weatherData <- tibble(
  date = seq.Date(from = as.Date("2023-01-01"), to = as.Date("2023-03-31"), by
  = "day"),  temp = round(runif(90, min = 15, max = 35), 1),
  prec = round(runif(90, min = 0, max = 20), 1),
  wind = round(runif(90, min = 5, max = 25), 1),
  humi = round(runif(90, min = 30, max = 90), 1),
  rain = ifelse(runif(90) > 0.7, "Yes", "No")
)

# Take Note on the Settings to simulate a Weather Pattern
# temperature is set to be between 15°C and 35°C
# precipitation is measured in millimeter [the amount of rain]
# rain is set to 30% chance to occur
# wind Speed is measure in km/h
# humidity is in Percent form
```

Part I: Predict Rain Using Weather Data

Step 1: Data Exploration

1. Inspect the dataset [use any command to describe the Dataset].

```
> dim(weatherData)
> ncol(weatherData)
> nrow(weatherData)
> str(weatherData)
> summary(weatherData)
> glimpse(weatherData)
> head(weatherData)
> tail(weatherData)
```

2. Check the dataset and filter for rainy and non-rainy days.

[pipe operator will be very useful]

[created dataset rainyDays w/ rain = 'Yes' and non_rainyDays when 'No']

```
> rainyDays <- weatherData%>%filter(rain == "Yes")
> print(rainyDays, n = 30)
```

```
> non_rainyDays <- weatherData%>%filter(rain == "No")
> print(non_rainyDays, n = 70)
```

3. Calculate descriptive statistics.

[use group_by(rain) then summarize(avgTemp = mean(temp), avgHum = mean(humi), numDays = n())]

```
> weatherDescStats <-
weatherData%>%group_by(rain)%>%summarize(avgTemp
= mean(temp),
                 avgHumi = mean(humi),
                 numDays = n())

> print(weatherDescStats)
```

What have you observed?

Discuss your findings below:

By analyzing the descriptive statistics result, the total number of days is 90, with 64 days being non-rainy (labeled as "No") and 26 days being rainy (labeled as "Yes"). The rain probability is approximately 28.9%, which aligns with the original data generation where rain was set to have a 30% chance of occurrence.

Regarding temperature, the average temperature for non-rainy days is 25.0°C, while for rainy days, the average temperature is slightly higher at 25.5°C. The minimal temperature difference suggests that temperature alone might not be a strong predictor of rain.

In terms of humidity, the average humidity for non-rainy days is 56.3%, while for rainy days, the average humidity is higher at 64.3%. This indicates a potential correlation between higher humidity and rainfall. So my essential observation is that there is a noticeable increase in humidity during rainy days. Additionally, the temperature remains relatively consistent across rainy and non-rainy days. The data supports the initial predictive rule that higher humidity (>70%) might be associated with rain.

Therefore, I would recommend that when planning the outdoor event, pay close attention to humidity levels. If humidity approaches or exceeds 70%, consider having a contingency plan for indoor activities or rescheduling.

Step 2: Data Cleanup and Data Wrangling

1. Check if there are missing values and do some cleanup.

[use the is.na(), replace_na() from tidyr with na.rm = TRUE].

```
> sum(is.na(weatherData))

> weatherData <- weatherData %>%
  replace_na(list(
    temp = mean(weatherData$temp, na.rm = TRUE),
    prec = 0,
    wind = mean(weatherData$wind, na.rm = TRUE),
    humi = mean(weatherData$humi, na.rm = TRUE),
    rain = "No"
  ))
```

Is there a missing values in the dataset? NO

If there is, then run the script below to rectify the problem:

```
weatherData <- weatherData %>%
  replace_na(list(
    temp = mean(weatherData$temp, na.rm = TRUE),
    prec = 0,
    wind = mean(weatherData$wind, na.rm = TRUE),
    humi = mean(weatherData$humi, na.rm = TRUE),
    rain = 'No'))
```

2. Create temperature categories.

[Low (< 20), Medium (between 20 to 30) and High (> 30)]

[use the mutate() and name the column to tempCat then use the case_when() function to apply the require conditions]

```
> weatherData <- weatherData%>%mutate(tempCat =
case_when(
                                  temp < 20 ~ "Low",
                                  temp >= 20 & temp <= 30 ~ "Medium",
                                  temp > 30 ~ "High"
                                 ))
```

Step 3: Visualize the Dataset

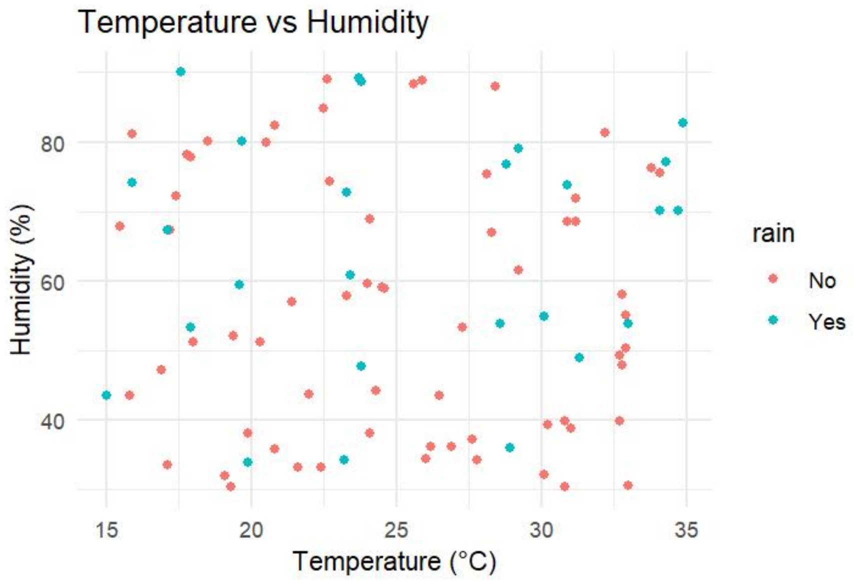
1. Create a scatter plot of between temperature and humidity.

[use the ggplot2 library and set aes(x=temp, y=humi, color=rain] [use geom_point() to specify that you want to use scatter plot]

```
> library(ggplot2)

> scatterPlot <- ggplot(weatherData, aes(x = temp, y = humi,
color = rain)) +
  geom_point() +
  labs(title = "Temperature vs Humidity",
       x = "Temperature (°C)",
       y = "Humidity (%)") +
  theme_minimal()

> print(scatterPlot)
```



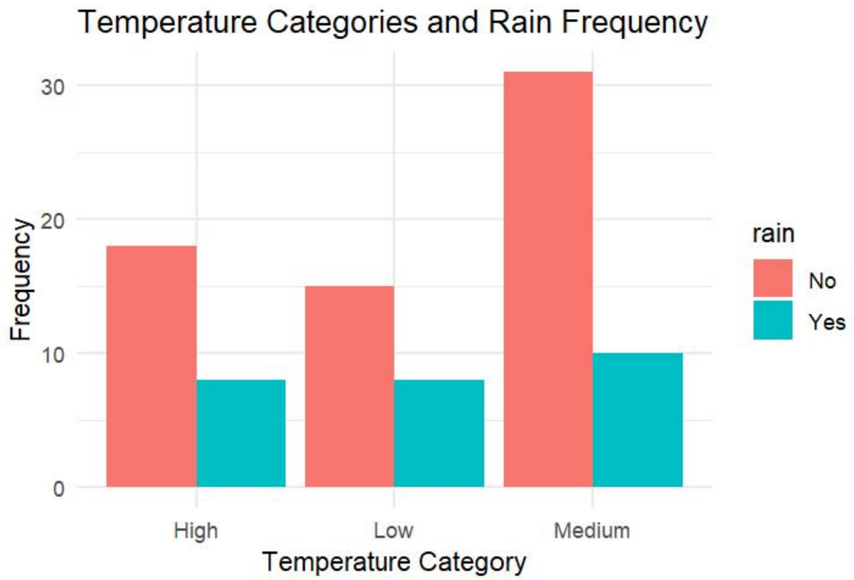
2. Create a bar chart of temperature categories [Low, Medium, High] against rain frequency.

[set aes(x = tempCat, fill = rain]

[use geom_bar(position = ‘dodge’) to specify the you want a bar graph]

```
>barChart <- ggplot(weatherData, aes(x = tempCat, fill =
rain)) +
  geom_bar(position = "dodge") +
  labs(title = "Temperature Categories and Rain Frequency",
       x = "Temperature Category",
       y = "Frequency") +
  theme_minimal()

> print(barChart)
```

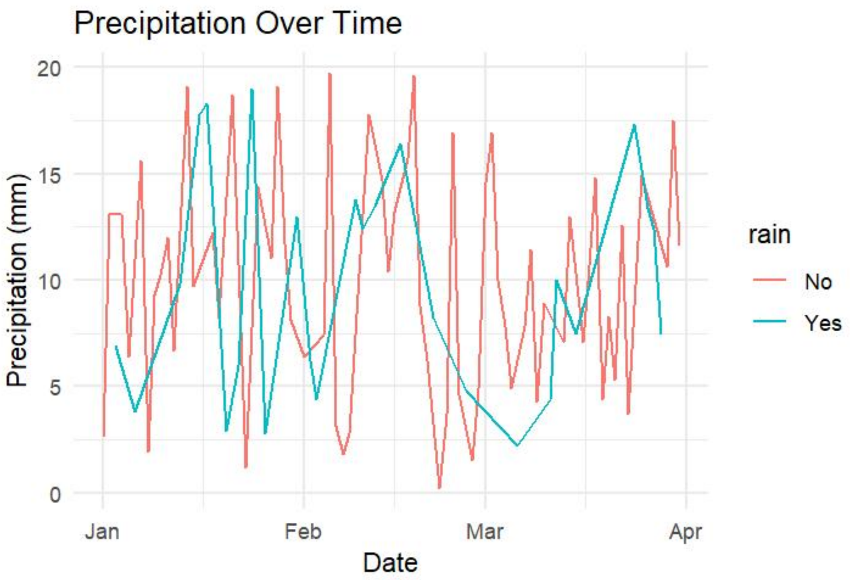


3. Create a line graph bar of precipitation [prec] over time [the whole duration of time recorded in the dataset. [set aes(x = date, y = prec, color = rain]

[use geom_line() to specify the you want a line graph]

```
> lineGraphBar <- ggplot(weatherData, aes(x = date, y =
prec, color = rain)) +
  geom_line() +
  labs(title = "Precipitation Over Time",
       x = "Date",
       y = "Precipitation (mm)") +
  theme_minimal()

> print(lineGraphBar)
```



Step 4: Predictive Analysis

- Identify patterns and create predictive rules.

i. If hum > 70% and temp < 25°C, rain will likely occur, use the filter() and save this dataset as rainPrediction.

```
> rainPrediction <- weatherData%>%filter(humi > 70, temp < 25)
> print(nrow(rainPrediction))
> print(rainPrediction)
```

How many occurrences where listed to meet the conditions above?
There are 16 occurrences listed that meets the condition above.

ii. with the rainPrediction dataset, compare it the original dataset [weatherData] and check the number of occurrences the matched with the rainPrediction dataset, you can name it as matchedPrediction.

```
> matchedPrediction <- weatherData%>%filter(humi > 70,
temp < 25 & rain == 'Yes')
> print(nrow(matchedPrediction))
> print(matchedPrediction)
```

How many predictions where correct?

There are 6 out of 16 predictions that were correct.

Have you noticed something in step 4? Take a look at the predictive rules, why was it set that way? What do you think?

In Step 4, we created a rule to predict rain based on certain levels of humidity and temperature. We chose these levels because it's commonly known that high humidity and lower temperatures can lead to rain. However, it's important to remember that many factors affect the weather. While these rules give us a basic idea, they might not cover all the complexities of weather patterns.

Another thing, what would be the best command(s) to at least arrive to the values closer to the predictive rules set in Step 4.1?

The predictive rule was created because we know that high humidity usually leads to rain, especially when temperatures are low. Warm air can hold more moisture, and when it cools down, that moisture can turn into rain. However, this rule is quite simple and doesn't consider other important factors like air pressure, wind patterns, and past weather conditions.

With all these considerations, and based on your own opinion in relation to the provided dataset, what are the best parameters [variables] to take into account in predicting the possibility of Rain to occur in the future?

In relation to the provided dataset, the predictive rule shows that high humidity and low temperature are strong indicators of rain. So the best parameters (variables) to consider for predicting the possibility of rain in the future could include weather parameters like **humidity**, **precipitation**, and **temperature,** which are crucial for rain prediction.

Part II: From Predictive to Prescriptive Analysis

What is Prescriptive Analytics?

- While *Predictive Analytics* **helps** us f**orecast what might happen** [predicting rain] *Prescriptive Analytics,* on the hand goes a step further by **offering recommendations on what actions to take based on these predictions**.
- It answers the question: "Given the predicted outcome, what should we do next?"

Applying Prescriptive Analytics to the Weather Dataset

Scenario Extension:

You now have a **predictive rule**: *"If Humidity > 70% and Temperature < 25°C, then, rain is likely to happen."*

Your next task is to decide whether to proceed with the outdoor event or implement a contingency plan based on the *predictive rule* to guide you on your actions.

Step 1: Define Actionable Scenarios

- Using these condition, you can create a subset of the weather dataset to be the main options [possible **actions]**.

Options	**Prediction**	**Action**
1	No Rain Predicted	Proceed with the Outdoor Activity
2	Rain Predicted	Move to an indoor venue or reschedule the Activity

Step 2: Implement the Classification base on the Rule.

- Use the predictive rule to classify days into actionable scenarios and then save it a modified dataset.

```
classified_WD <- weatherData %>%
 mutate(
  action = case_when(
   humi > 70 & temp < 25 ~ "Indoor or Postpone",
   TRUE ~ "Proceed with Outdoor Activity"
  )
 )
```

You can now view the modified dataset with the added action column to get an idea how these proposed Rule allows you to make a calculated decision in the future.

What have you observed? Write it down below. You can ran scripts to further describe what you see.

```
> head(classified_WD)

> tail(classified_WD)
```

```
# Summary of actions

> table(classified_WD$action)


summary_actions <- classified_WD %>%

  group_by(action) %>%

  summarize(

    avg_temperature = mean(temp),

    avg_humidity = mean(humi),

    avg_precipitation = mean(prec),

    total_days = n()

  )

> print(summary_actions)


# Percentage of days for each action

action_percentage <- classified_WD %>%

  count(action) %>%

  mutate(percentage = n / sum(n) * 100)

> print(action_percentage)
```

In our weather analysis, we created a simple way to decide if our outdoor event can happen. We made a rule that says if humidity is over 70% and temperature is below 25°C, we should move the event inside or reschedule. Looking at our data, most days - 74 out of 90 - are good for an outdoor event. Only 16 days might need us to change our plans. This means our rule helps us be careful without stopping the event completely.

The classification system works like a quick weather checker. It looks at humidity and temperature to tell us if a day is safe for an outdoor activity. This helps event planners make fast decisions and have a backup plan ready. Our method shows that with just two simple weather measurements, we can get a good idea of whether an event can happen outside. It's not perfect, but it gives us a helpful guide. The key is to stay flexible and be ready to change plans if the weather doesn't cooperate.

By using this simple rule, we can feel more confident about planning our outdoor charity event. We know most days will be fine, and we have a plan for the few days that might be tricky. This approach takes some of the stress out of event planning by giving us a clear way to make decisions based on weather conditions.

Step 3: Summarize the Actions.

- Evaluate the distribution of the recommended actions.

```
summed_WD <- classified_WD %>%
  group_by(action) %>%
  summarize(days = n()) %>%
  arrange(desc(days))
```

What have you observed? Write it down below.

Based on the weather prediction model, we set a rule to move indoors or reschedule when humidity is high (over 70%) and temperature is low (below 25 °C). This careful approach helps protect our outdoor event from bad weather. The data shows that most days (74 out of 90) are good for an outdoor event. This is great news for our event planning and gives us confidence that we can host the charity event outside. But we shouldn't ignore the 16 days that might have weather problems. Event organizers should:

- Create a backup plan
- Find an indoor location we can use
- Think about changing dates for the days with high weather risks

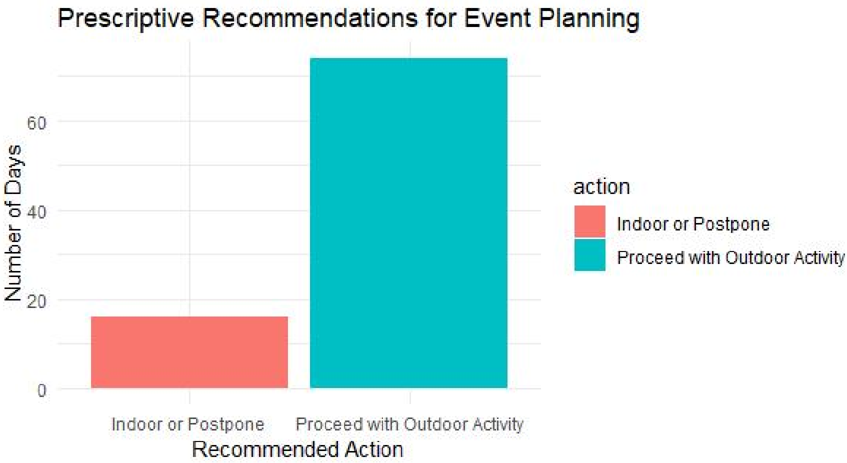
Since only 17.8% of days might need an indoor option, we can plan our event with confidence. We just need to be ready to change our plans if needed. This study shows how using data can help us make smarter decisions about event planning. By looking at weather information carefully, we can reduce uncertainty and be better prepared.

Step 4: Visualize the Recommendation.

- Use a bar chart to display the count of days for each action:

```
ggplot(summed_WD, aes(x = action, y = days, fill = action)) +
  geom_bar(stat = 'identity') +
  labs(title = "Prescriptive Recommendations for Event Planning",
       x = "Recommended Action",
       y = "Number of Days") +
  theme_minimal()
```

What have you observed? Write it down below.



The bar chart helps us understand the weather conditions for our outdoor charity event. Most of the days - exactly 74 out of 90 - are good for an outdoor activity. Only 16 days might need us to change our plans and move inside or reschedule. The chart clearly shows two bars with different heights. The taller bar represents the days we can safely hold the event outside, which is the majority. The shorter bar shows the days when we might need to be careful about the event's location. This means we have a very good chance of hosting the charity event outdoors. About 82% of the days are perfect for the plans. The small number of risky days - around 18% - doesn't mean we should cancel the event. Instead, it tells us we should have a backup plan.

I recommend going ahead with the outdoor event. But be smart and prepare an indoor option just in case. Having a flexible plan will help us handle the few days that might have tricky weather. This way, we can make sure the charity event happens, rain or shine. The chart shows us that data can help make better decisions. By looking carefully at the weather information, we can plan our event with confidence and be ready for any small challenges that might come our way.

ACTIVITY REALIZATION

In first part of the activity, you have preformed the three types of Data Analytics [*descriptive*, *diagnostic* and *predictive*] all in just four steps. By following these pattern, you will be able to master the basic process in Exploratory Data Analysis [**EDA**]. Also, the key takeaway in predicting an event to occur is by taking a closer look at the dataset and its graphs, identify the parameters and do some tests and validation to check its effectiveness as demonstrated in Step 4.

On the second part, you were introduced to the fourth type of Analytics [*Prescriptive Analytics*] where you used the information you have in **Part I** [from *Descriptive* to *Predictive*] and leverage those information to draw out two *"actionable"* possibilities for the organization to decide in consideration with the upcoming outdoor activity.

Another method to draw out predictions and prescriptions based on a given dataset is to use the Linear Regression Model, which will be covered in a hands-on demonstration, so please stay tuned.

~END~