Weather\_datathon

Erick onyango ooko

**Temperature Trends**

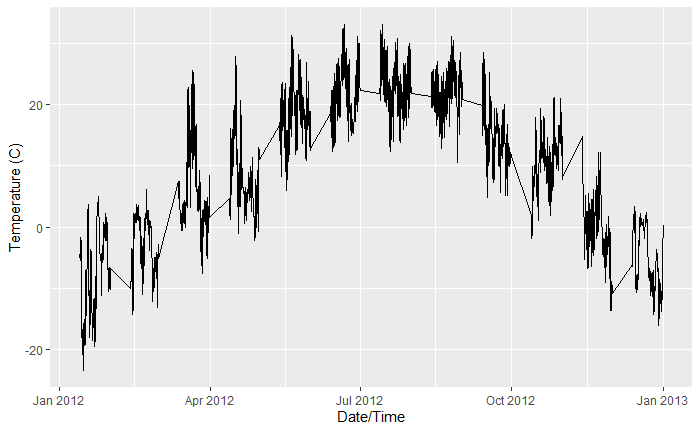


Figure 1. Temperature Trends over Time

According to Figure1, there was a steady temperature change over the year, from lowest temperatures in January 2012 to a peak in June. It then fell to lower temperatures by January 2013.

**The relationship between wind speed and temperature**

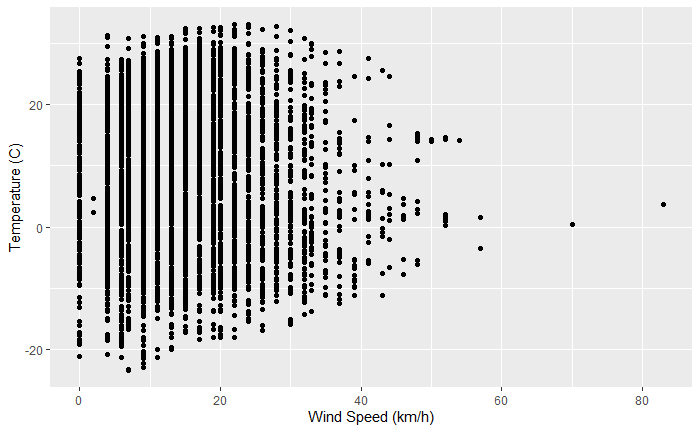


Figure 2. Relationship between wind speed and Temperature

According to figure 2, there is a negative weaker relationship between windspeed and temperature.

**Forecasting temperature using ARIMA**

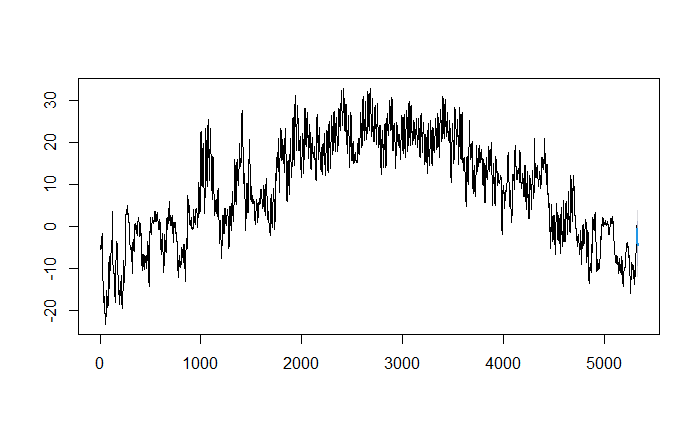


Figure 3. ARIMA (5, 1, 1) forecast of temperature

Figure 3 showed a forecast of temperature for 10 data points horizon. It illustrated an expected increase in temperature for the period.

**Detecting outliers or anomalies**

## Figure 4. Boxplot of Temperature

According to figure 4, all the data points lie close to the mean hence outlliers in the dataset.

**Decompose the Data**

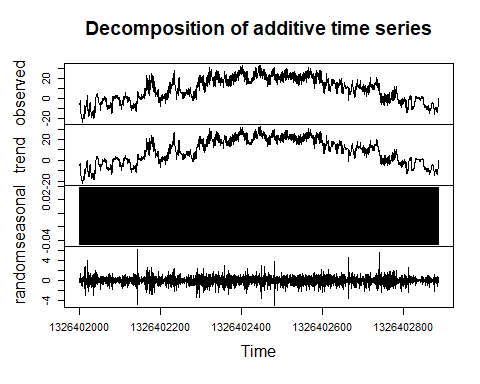


Figure 5. Decomposition of temperature data

According to figure 5, the dataset is stationary since there is no significant trend. Besides, the seasonal trend shows a consistent and stationary seasonal pattern in the data

**The correlation matrix**

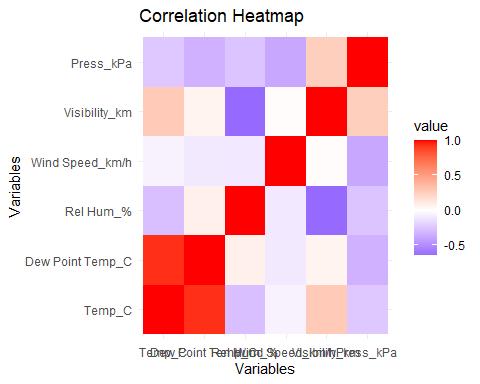


Figure 6. Correlation Heatmap of variables

According to figure 6, there is a strong correlation between temperature and dew point temperature, a weaker correlation between temperature and visibility, and negative weaker correlation between temperature and pressure, wind speed, and humidity.

**Appendix**

# Load the dataset  
library(readxl)  
weather\_data <- read\_excel("C:/Users/admin/Downloads/weather data.xlsx")  
  
View(weather\_data)

## Temperature trends over time

library(ggplot2)  
  
ggplot(weather\_data, aes(x = as.POSIXct(`Date/Time`, format = "%m/%d/%Y %H:%M"), y = Temp\_C)) +  
 geom\_line() +  
 xlab("Date/Time") +  
 ylab("Temperature (C)") +  
 ggtitle("Temperature Trends Over Time")

## Warning: Removed 3456 rows containing missing values (`geom\_line()`).

## Analyze the relationship between wind speed and temperature

ggplot(weather\_data, aes(x = `Wind Speed\_km/h`, y = Temp\_C)) +  
 geom\_point() +  
 xlab("Wind Speed (km/h)") +  
 ylab("Temperature (C)") +  
 ggtitle("Relationship between Wind Speed and Temperature")

correlation <- cor(weather\_data$`Wind Speed\_km/h`, weather\_data$Temp\_C)  
correlation

## [1] -0.061876

## Forecasting temperature using ARIMA

library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

# Assuming the temperature data is in a column named "Temp\_C"  
temperature <- ts(weather\_data$Temp\_C)  
  
# Fit ARIMA model  
model <- auto.arima(temperature)  
  
# Forecast next 10 time points  
forecast <- forecast(model, h = 10)  
  
# Plot the forecast  
plot(forecast)

## Detecting outliers or anomalies

boxplot(weather\_data$Temp\_C, main = "Temperature Boxplot")

## Decompose the Data

library(stats)  
# Convert 'Years' column to a date format  
weather\_data$`Date/Time` = as.POSIXct(weather\_data$`Date/Time`, format = "%m/%d/%Y %H:%M")  
  
  
# Remove rows with null values  
clean\_data <- weather\_data[complete.cases(weather\_data), ]  
weather\_data = clean\_data  
  
# Convert the data to a time series object  
ts\_data <- ts(weather\_data$Temp\_C, frequency = 6, start = c(min(weather\_data$`Date/Time`)))  
  
#decompose the ts  
decomposed\_data<-decompose(ts\_data)  
  
plot(decomposed\_data)

## Calculate the correlation matrix

# Select only the numeric columns  
numeric\_columns <- weather\_data[, sapply(weather\_data, is.numeric)]  
  
# Calculate the correlation matrix for numeric columns  
cor\_matrix <- cor(numeric\_columns)  
  
# Print the correlation matrix  
print(cor\_matrix)

library(ggplot2)  
  
# Convert correlation matrix to long format  
cor\_df <- reshape2::melt(cor\_matrix)  
  
# Plot the heatmap using ggplot2  
ggplot(data = cor\_df, aes(x = Var2, y = Var1, fill = value)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +  
 labs(title = "Correlation Heatmap", x = "Variables", y = "Variables") +  
 theme\_minimal()

# Add color legend  
heat\_colors <- c("blue", "white", "red")  
breaks <- seq(-1, 1, by = 0.2)  
labels <- c("Low Correlation", "", "", "", "", "High Correlation")