PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

BY

TIMOTHY A. OLATUNJI

Windspeed is an essential factor in agriculture, government policies, flight operations and safety. High windspeed causes delays at takeoff and landing, since nearly every airplane encounters high windspeed during its ascending or descending. Windspeed of more than 30-35 kts (approximately 34-40 mph) generally prevent take-off and landing (Schrader, 2023).

This portfolio work uses a time series data collected every 3 hours from May 1 2018 00:00 to May 31 2018 21:00. Various parameters were collected in every longitude and latitude in the United Kingdom.

Table 1.0 Parameters in the dataset

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

Parameter	Description	Measuring Unit
XLAT	Latitude	
XLONG	Longitude	
TSK	Skin temperature or surface temperature	oK (Kelvin)
PSFC	Surface pressure	Pa (Pascal)
U10	X component of wind at 10m	m/s
V10	Y component of wind at 10m	m/s
Q2	2- meter specific humidity	Kg/Kg
Rainc	Convective rain (Accumulated precipitation)	mm
Rainnc	Non-convective rain	Mm
Snow	Snow water equivalent	Kg/m2
TSLB	Soil temperature	oK
SMOIS	Soil Moisture	m3/m3

One fifty-two Longitude and Latitude above and below MIA (Middlesbrough International Airport) Longitude and Latitude were selected to perform Exploratory data analysis on.

Linear Interpolation was used on the dataset to handle NA/Null values. Interpolation finds average of immediate value above and beneath Null values. For NA/Null values at the beginning/ end of the dataset, Interpolation would not be able to handle them, hence they were deleted.

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

Outliers were calculated using Zscore and handled using winsorization. Outliers handling is as important as handling NA/Null values in Data Analysis as it helps remove “noise” which are called extreme values which might have been inputted due to computational/data collection errors.

After this EDA, MIA longitude and latitude data point was extracted. The X and Y components of wind were extracted to calculate windspeed. After which Statistical model (ARIMA) and three machine learning models (Linear Regression, Support Vector Regression and Random Forest) were used to forecast/predict real time windspeed at MIA. It was detected that Random Forest with ntrees 500 performed best out of all the models. It is then safe to say windspeed at MIA is better predicted using ML than Forecasted using Statistical methods.

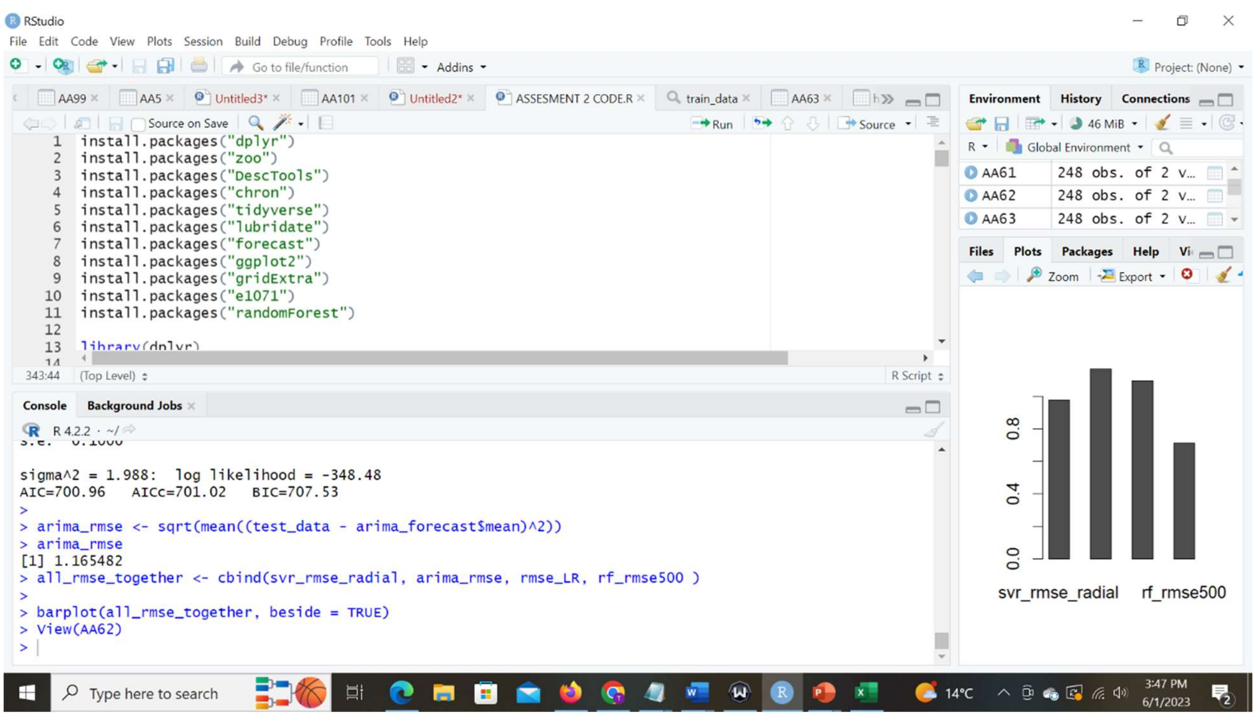

Fig 1.0 Installing necessary R libraries

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

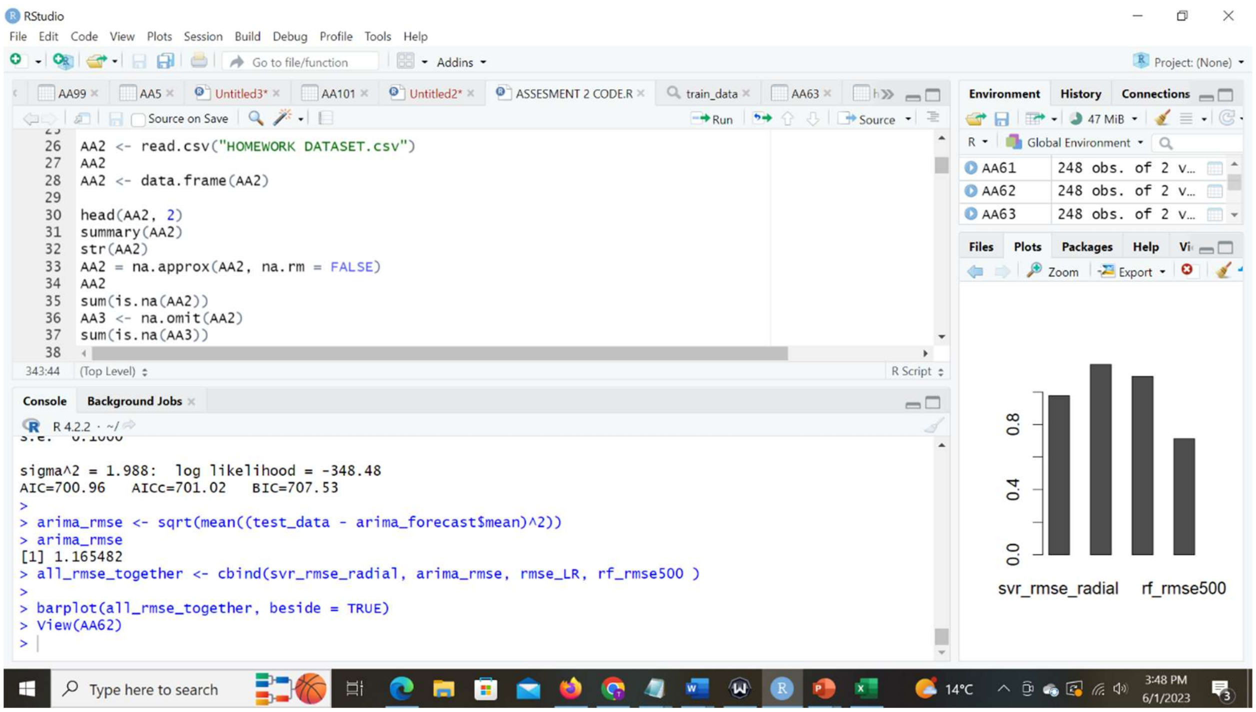

Fig 1.1 Loading dataset and performing Linear Interpolation

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

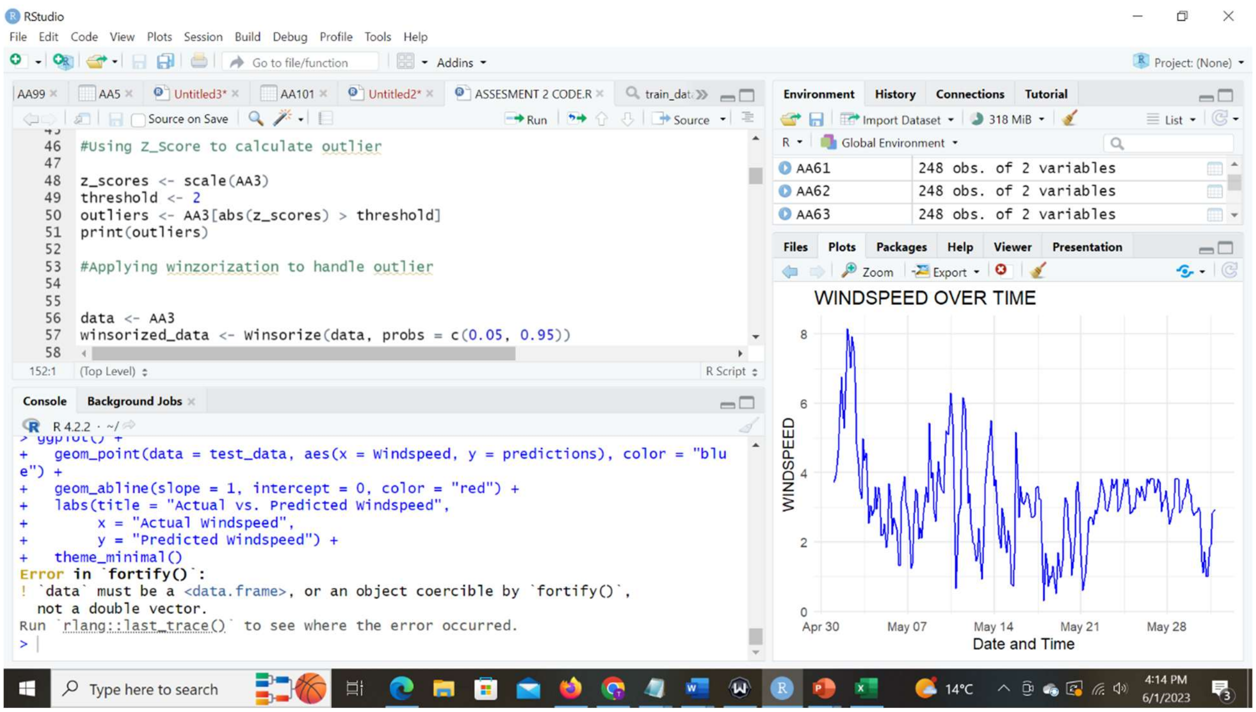

Fig 1.2 Using Zscore to calculate outlier and Winsorization to handle it.

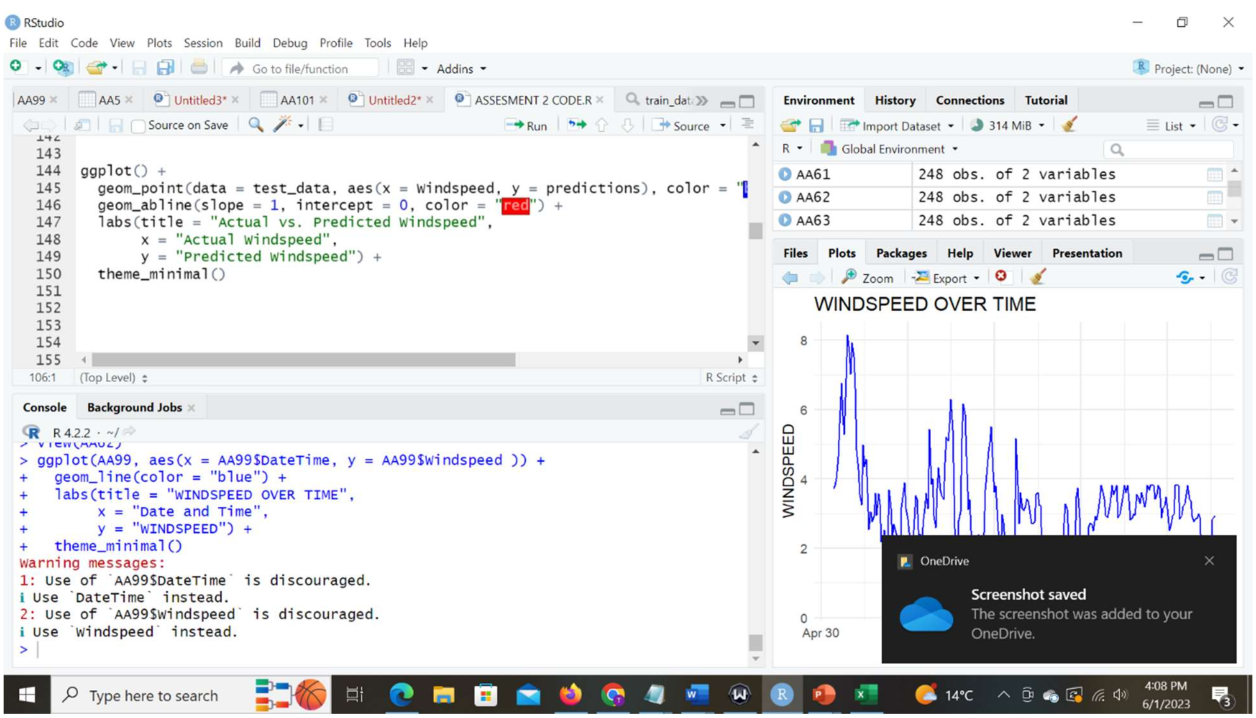

Fig 1.3 Plot of Windspeed over Time for MIA Longitude and Latitude.

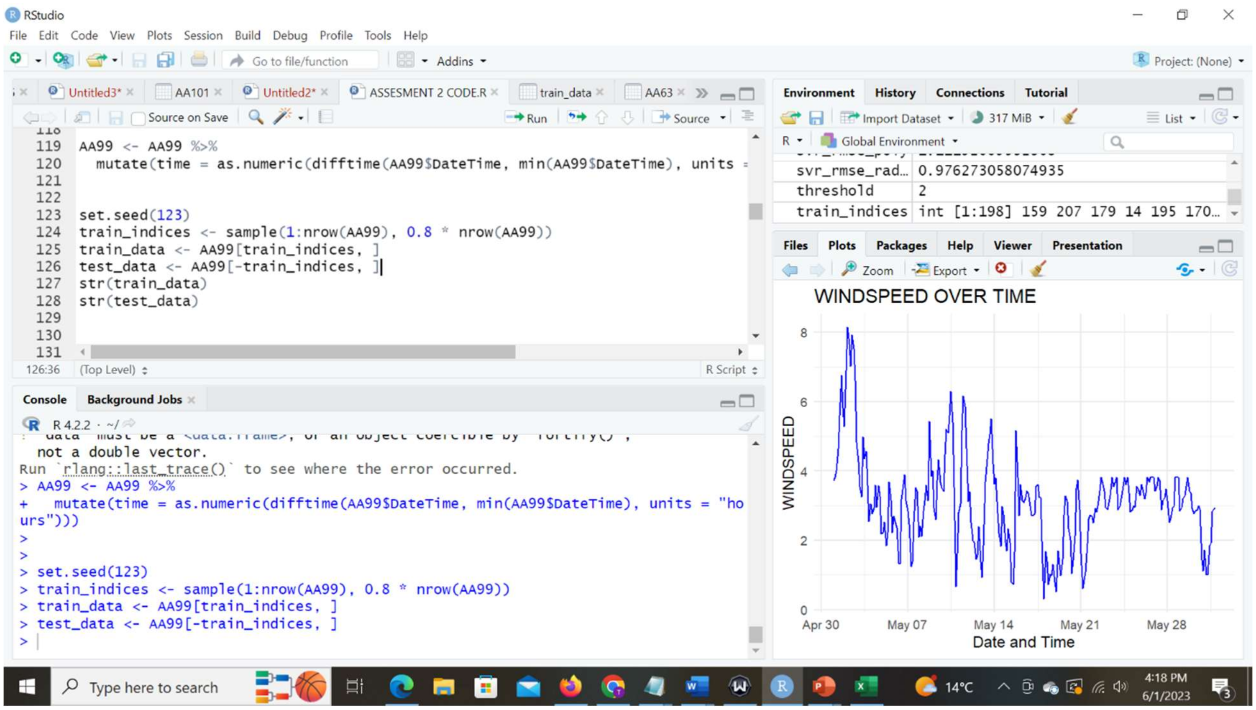

Fig 1.4 Splitting data into test and train

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

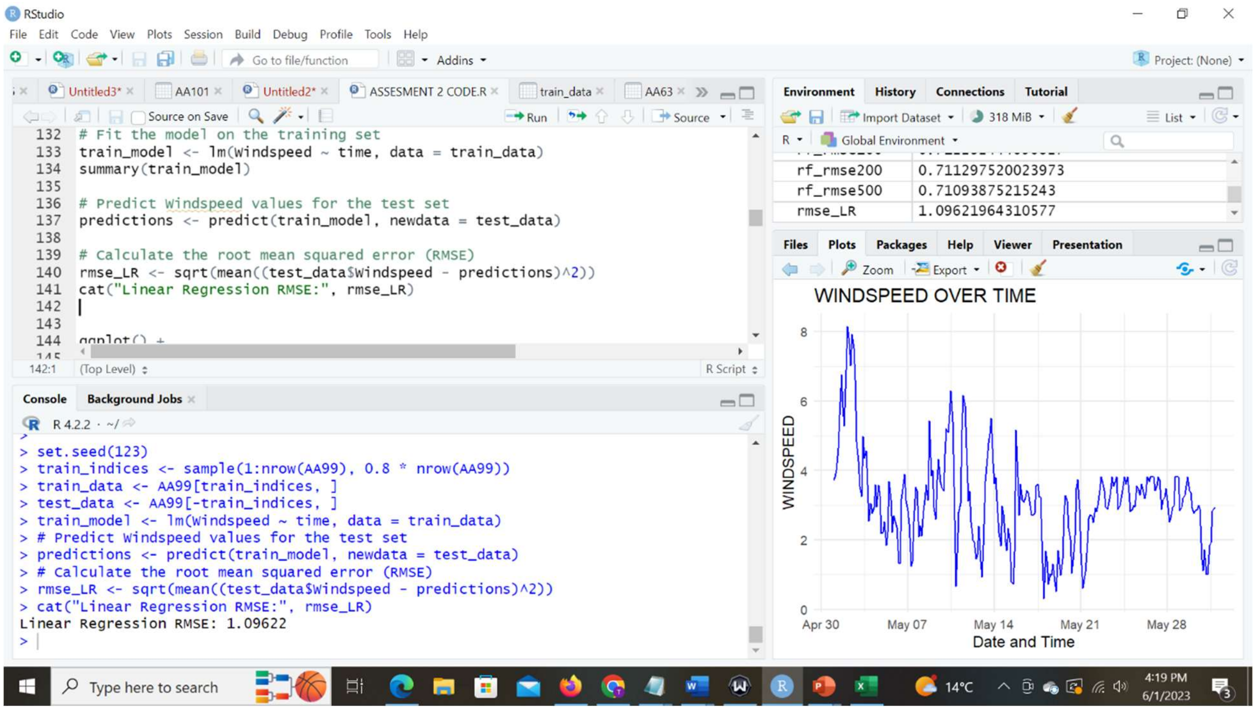

Fig 1.5 Fitting Linear Regression Model and using RMSE as Evaluation Matrice

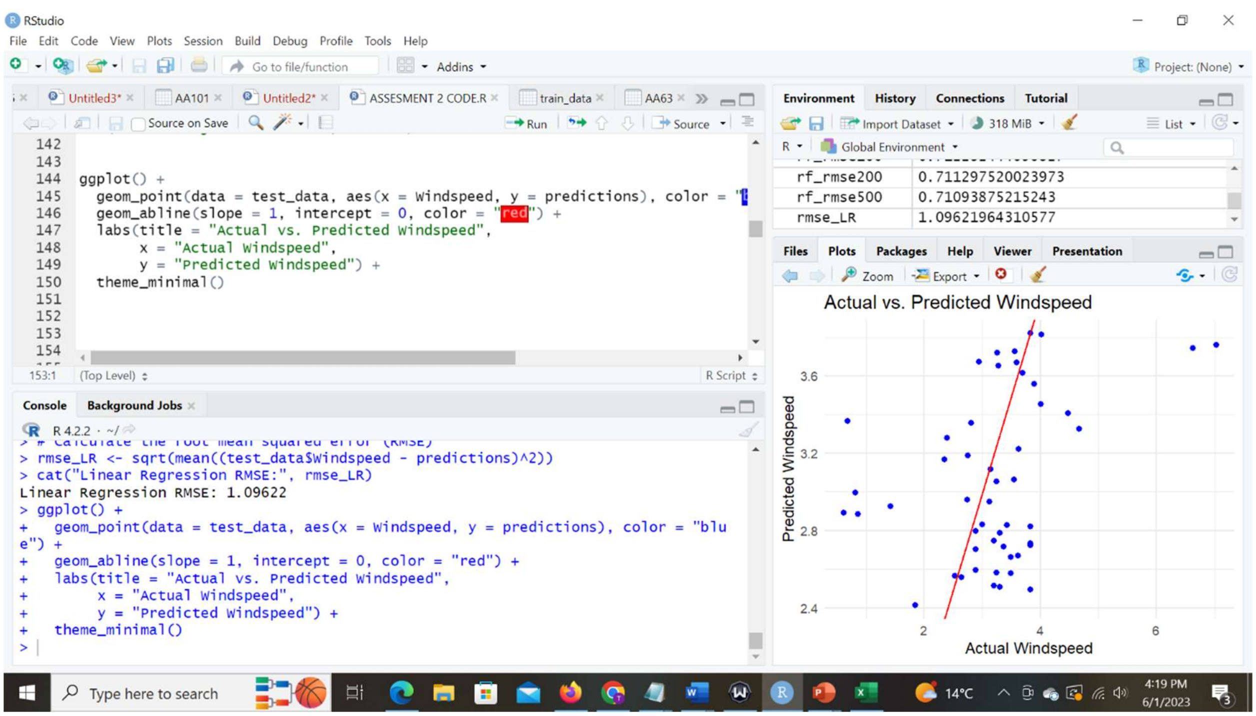

Fig 1.6 Plotting Actual and Predicted windspeed using LR.

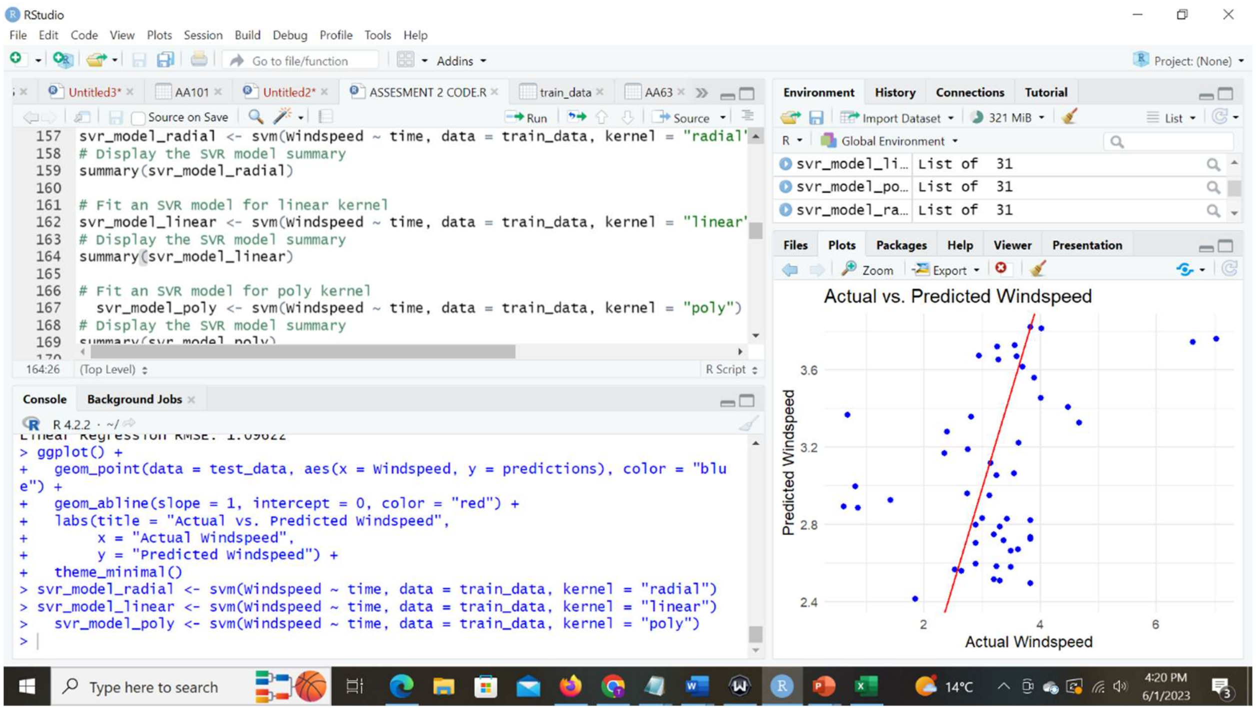

Fig 1.7 Fitting all SVR models on Linear, Poly and Radial kernels.

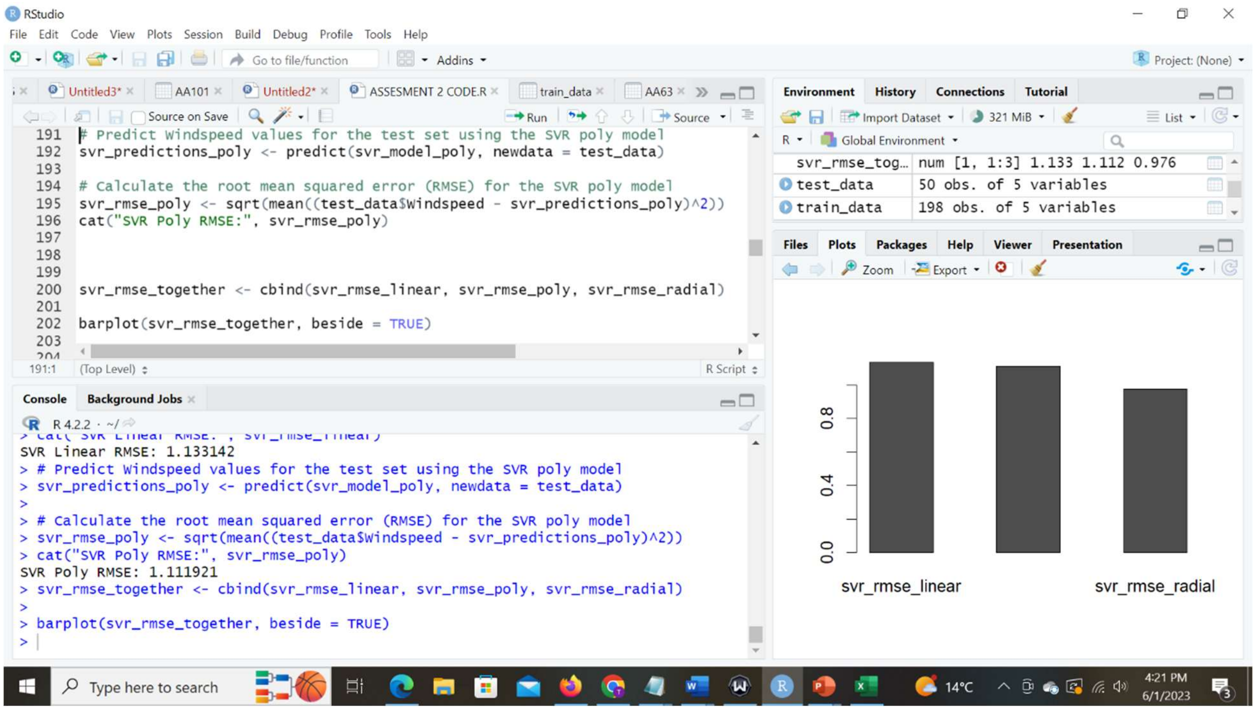

Fig 1.8 SVR radial kernel performs best and hence will be used to compare LR, RF and ARIMA

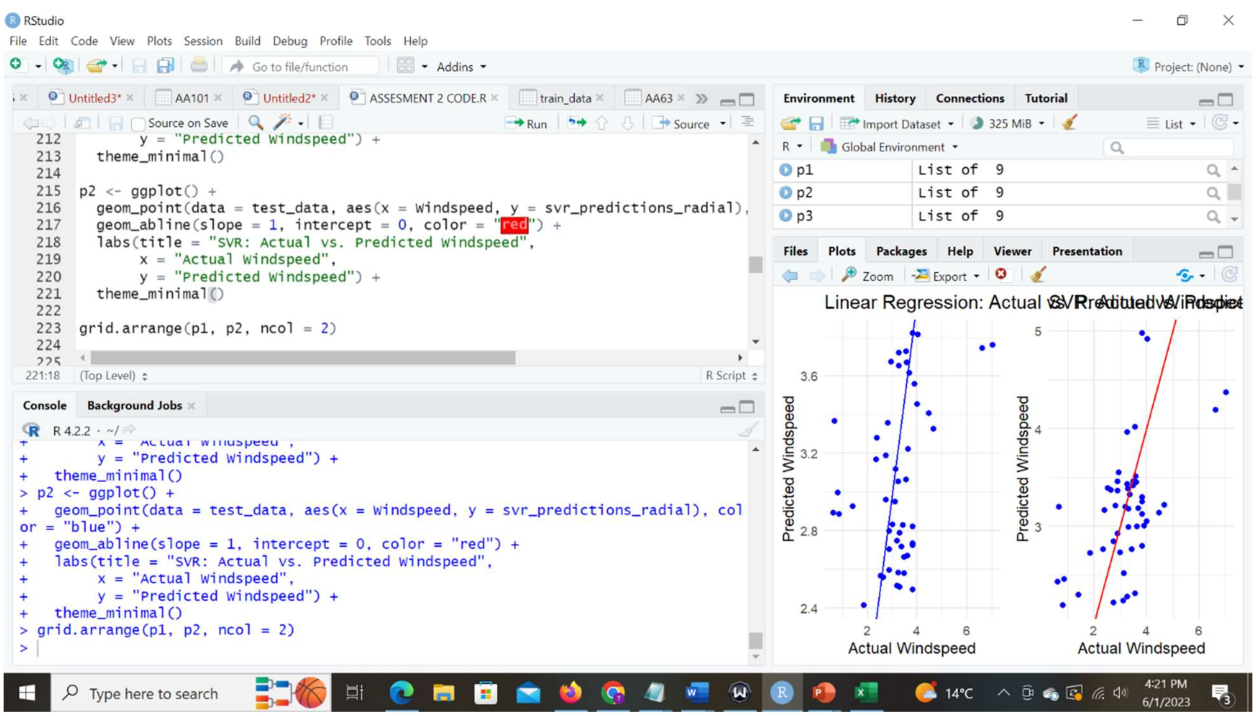

Fig 1.9 Plot of Actual and Predicted windspeed using LR and SVR Radial.

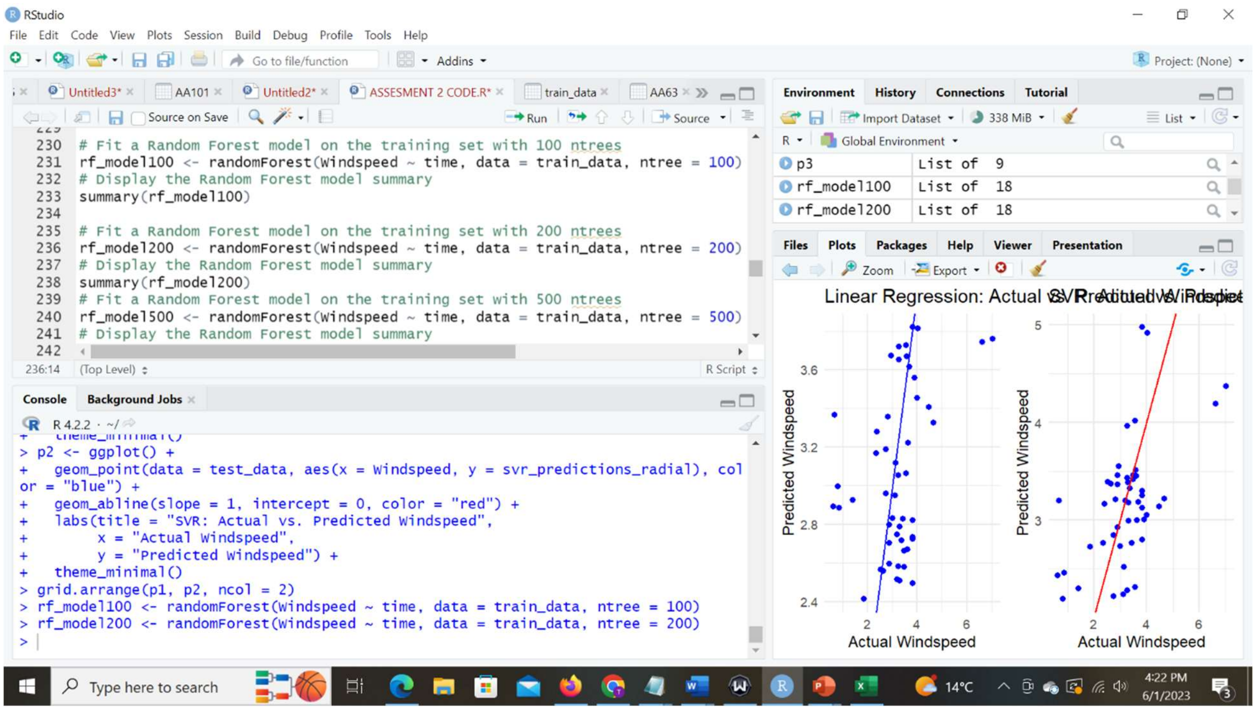

Fig 2.0 Fitting RF models on 100, 200 and 500 ntrees.

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

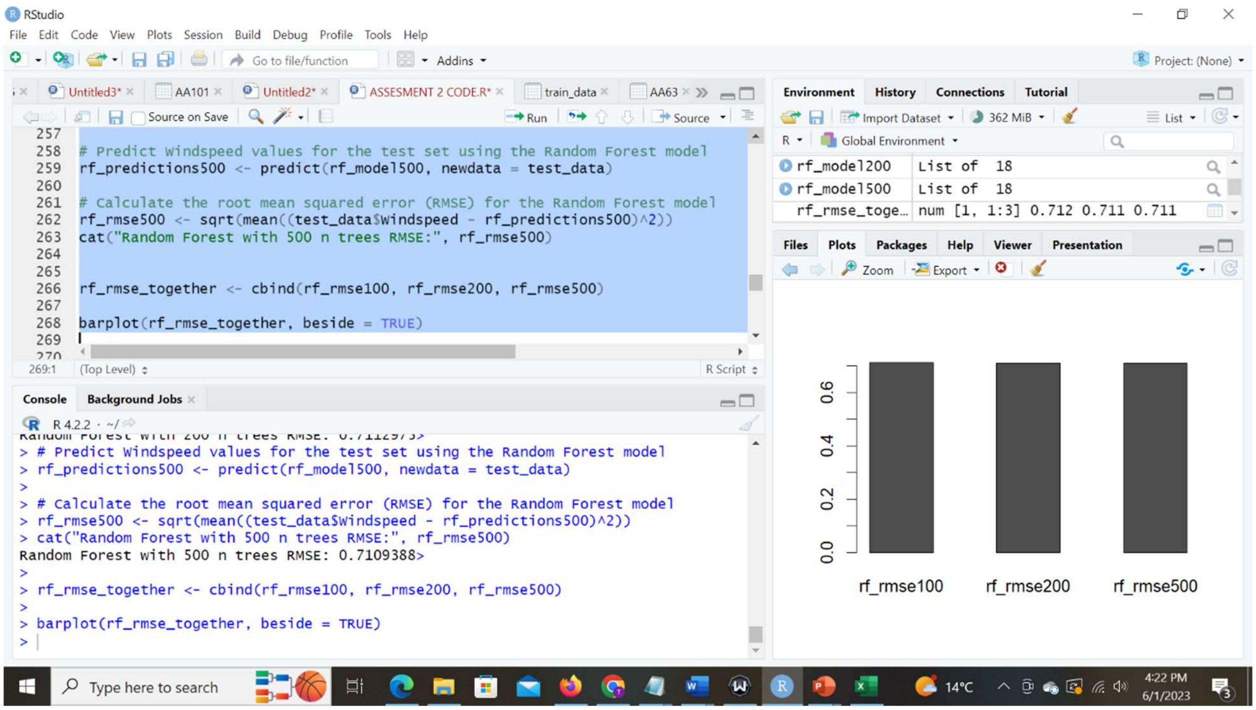

Fig 2.1 RF with 500 ntrees has the lowest RMSE hence performs best.

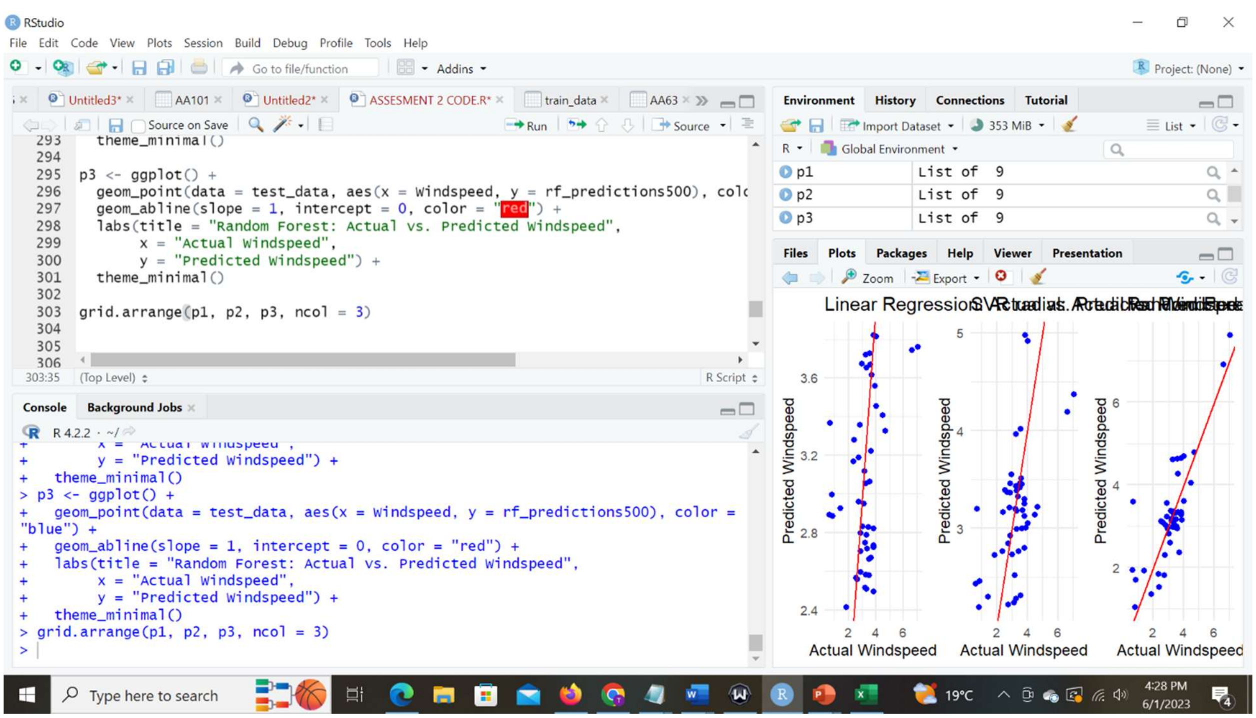

Fig 2.2 Plot of Actual and Predicted windspeed using LR, SVR Radial and RF with 500 ntrees

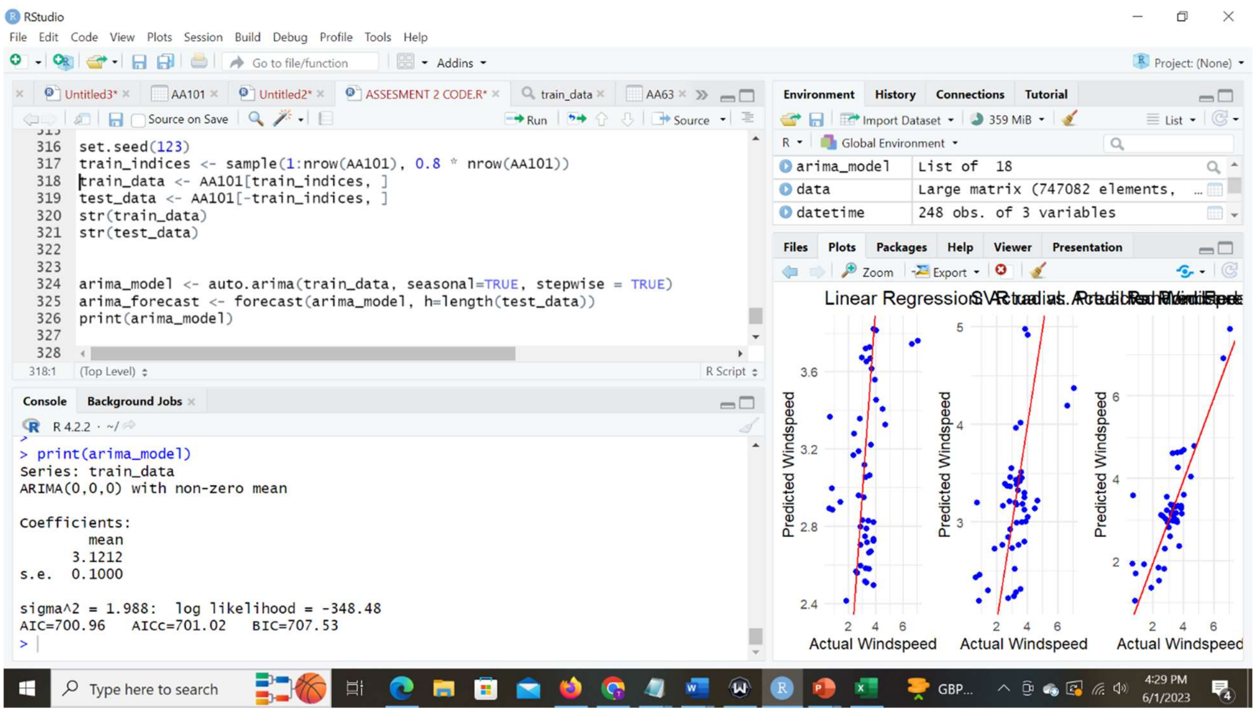

Fig 2.3 Splitting Training and Testing dataset for ARIMA model and fitting it.

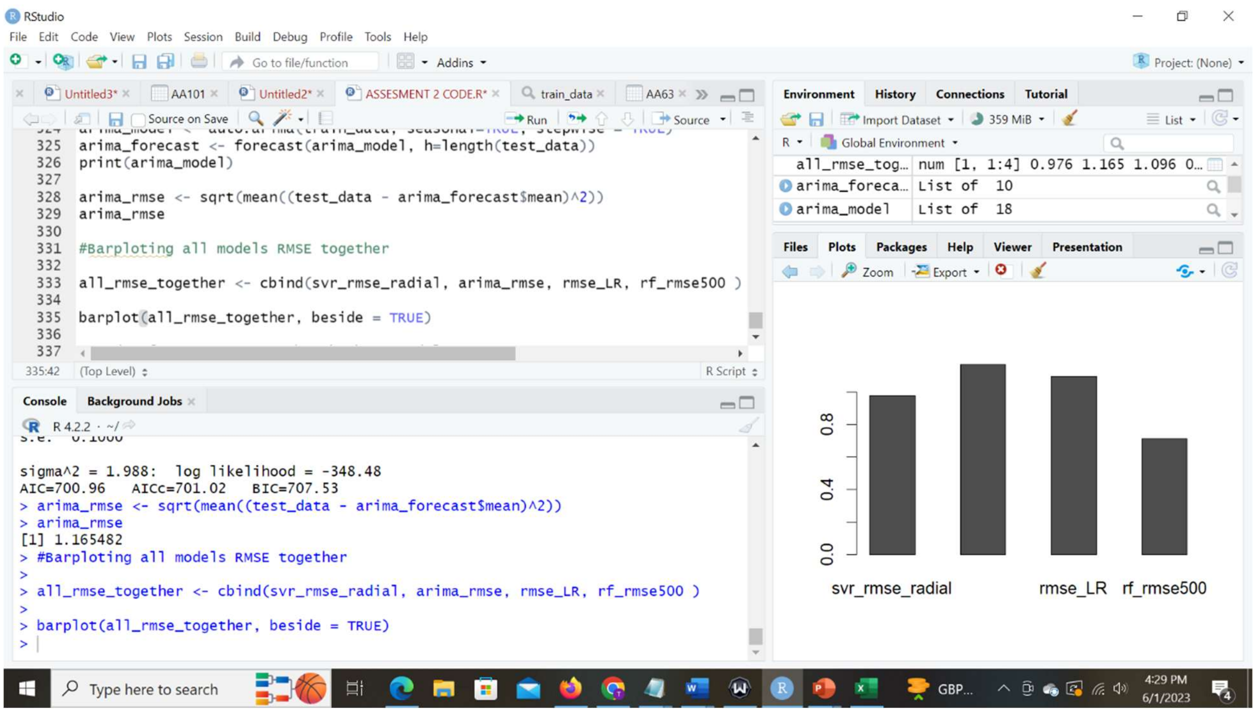

Fig 2.4 Barplot of all model RMSE.

This shows that RF with ntrees 500 performs better. Hence the Middlesbrough International airport aviation sector can deploy Random Forest with ntress 500 for their windspeed forecast. Other researchers/analyst can use other statistical models like prophet model, or machine learning models like decision tree to analyze the data set. The dataset is available on my Github.

Appendices (Full Rcode)

```
install.packages("dplyr")
install.packages("zoo")
install.packages("DescTools")
install.packages("chron")
install.packages("tidyverse")
install.packages("lubridate")
```

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
install.packages("forecast")
install.packages("ggplot2")
install.packages("gridExtra")
install.packages("e1071")
install.packages("randomForest")

library(dplyr)
library(zoo)
library(DescTools)
library(chron)
library(tidyverse)
library(lubridate)
library(forecast)
library(ggplot2)
library(gridExtra)
library(e1071)
library(randomForest)


AA2 <- read.csv("HOMEWORK DATASET.csv")
AA2
AA2 <- data.frame(AA2)

head(AA2, 2)
summary(AA2)
str(AA2)
AA2 = na.approx(AA2, na.rm = FALSE)
AA2
```

```
sum(is.na(AA2))

AA3 <- na.omit(AA2)

sum(is.na(AA3))

getwd()

write.csv(AA3, "C://Users//hp//Documents//CLEANEDDATA.csv", row.names=FALSE)


#Boxplot to visualize outliers

boxplot(AA3, main = "Boxplot", names = "Outliers")


plot(density(AA3))


#Using Z_Score to calculate outlier


z_scores <- scale(AA3)

threshold <- 2

outliers <- AA3[abs(z_scores) > threshold]

print(outliers)


#Applying winzorization to handle outlier



data <- AA3

winsorized_data <- Winsorize(data, probs = c(0.05, 0.95))

print(winsorized_data)


AA4 <- winsorized_data


write.csv(AA4, "C://Users//hp//Documents//CLEANEDDATA.csv", row.names=FALSE)
```

Timothy A Olatunji

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
AA601 <- as.data.frame(t(rep(c("TSK", "PSFC", "U10", "V10", "Q2", "RAINC", "RAINNC", "SNOW", "TSLB",
"SMOIS"), 248)))

AA601

write.csv(AA601,"C://Users//hp//OneDrive//Documents//CLEANED ROW.csv", row.names=FALSE)



AA5 <- read.csv("LOCATION POINTN.csv", header = FALSE)

AA5

AA6 <- as.data.frame(t(AA5))

AA61 <- AA6[is.element(AA6$V1, c('V10')),]

AA62 <- AA6[is.element(AA6$V1, c('U10')),]

AA62$V3 <- c(AA61$V2)

AA62 <- AA62[ , !names(AA62) %in%

          c("V1")]



AA63 <- AA62 %>% rename(V10 = V3, U10 = V2)


write.csv(AA63,"C://Users//hp//OneDrive//Documents//U10V10.csv", row.names=FALSE)


AA99 <- read.csv("U10V10.csv", header = TRUE)

summary(AA99)


AA99$Windspeed <- sqrt((AA99$U10)^2 + (AA99$V10)^2)


AA99
```

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
hn <- as.data.frame(seq(0,23, 3))

hm <- merge(hn, 0)


interval_3hours <- data.frame('INTERVAL' = chron(time = paste(hm$`seq(0, 23, 3)`, ':', hm$y, ':', 0)))

interval_3hours <- data.frame('INTERVAL' = interval_3hours[order(interval_3hours$INTERVAL), ])

yearss <- as.data.frame(seq.Date(from = as.Date('2018-05-01'), to = as.Date('2018-05-31'), by = 'days'))


datetime <- merge(yearss, chron(time = paste(hm$`seq(0, 23, 3)`, ':', hm$y, ':', 0)))

colnames(datetime) <- c('date', 'time')


# create datetime

datetime$dt <- as.POSIXct(paste(datetime$date, datetime$time))

# create right order

datetime <- datetime[order(datetime$dt), ]

row.names(datetime) <- NULL


AA99$DateTime <-  datetime$dt




ggplot(AA99, aes(x = AA99$DateTime, y = AA99$Windspeed )) +

  geom_line(color = "blue") +

  labs(title = "WINDSPEED OVER TIME",

       x = "Date and Time",

       y = "WINDSPEED") +

  theme_minimal()
```

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
AA99 <- AA99 %>%
  mutate(time = as.numeric(difftime(AA99$DateTime, min(AA99$DateTime), units = "hours")))



set.seed(123)
train_indices <- sample(1:nrow(AA99), 0.8 * nrow(AA99))
train_data <- AA99[train_indices, ]
test_data <- AA99[-train_indices, ]
str(train_data)
str(test_data)



# Fit the model on the training set
train_model <- lm(Windspeed ~ time, data = train_data)
summary(train_model)

# Predict Windspeed values for the test set
predictions <- predict(train_model, newdata = test_data)

# Calculate the root mean squared error (RMSE)
rmse_LR <- sqrt(mean((test_data$Windspeed - predictions)^2))
cat("Linear Regression RMSE:", rmse_LR)



ggplot() +
  geom_point(data = test_data, aes(x = Windspeed, y = predictions), color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red") +
```

```
 labs(title = "Actual vs. Predicted Windspeed",
     x = "Actual Windspeed",
     y = "Predicted Windspeed") +
 theme_minimal()
```

```
# Fit an SVR model on the training set
svr_model_radial <- svm(Windspeed ~ time, data = train_data, kernel = "radial")
# Display the SVR model summary
summary(svr_model_radial)

# Fit an SVR model for linear kernel
svr_model_linear <- svm(Windspeed ~ time, data = train_data, kernel = "linear")
# Display the SVR model summary
summary(svr_model_linear)

# Fit an SVR model for poly kernel
 svr_model_poly <- svm(Windspeed ~ time, data = train_data, kernel = "poly")
# Display the SVR model summary
summary(svr_model_poly)
```

```
# Predict Windspeed values for the test set using the SVR radial model
svr_predictions_radial <- predict(svr_model_radial, newdata = test_data)
```

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
# Calculate the root mean squared error (RMSE) for the SVR radial model
svr_rmse_radial <- sqrt(mean((test_data$Windspeed - svr_predictions_radial)^2))
cat("SVR radial RMSE:", svr_rmse_radial)


# Predict Windspeed values for the test set using the SVR linear model
svr_predictions_linear <- predict(svr_model_linear, newdata = test_data)

# Calculate the root mean squared error (RMSE) for the SVR linear model
svr_rmse_linear <- sqrt(mean((test_data$Windspeed - svr_predictions_linear)^2))
cat("SVR Linear RMSE:", svr_rmse_linear)


# Predict Windspeed values for the test set using the SVR poly model
svr_predictions_poly <- predict(svr_model_poly, newdata = test_data)

# Calculate the root mean squared error (RMSE) for the SVR poly model
svr_rmse_poly <- sqrt(mean((test_data$Windspeed - svr_predictions_poly)^2))
cat("SVR Poly RMSE:", svr_rmse_poly)


svr_rmse_together <- cbind(svr_rmse_linear, svr_rmse_poly, svr_rmse_radial)

barplot(svr_rmse_together, beside = TRUE)
```

```
# Plot the actual vs. predicted values for the linear regression and SVR radial models
p1 <- ggplot() +
  geom_point(data = test_data, aes(x = Windspeed, y = predictions), color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "blue") +
  labs(title = "Linear Regression: Actual vs. Predicted Windspeed",
       x = "Actual Windspeed",
       y = "Predicted Windspeed") +
  theme_minimal()

p2 <- ggplot() +
  geom_point(data = test_data, aes(x = Windspeed, y = svr_predictions_radial), color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "SVR: Actual vs. Predicted Windspeed",
       x = "Actual Windspeed",
       y = "Predicted Windspeed") +
  theme_minimal()

grid.arrange(p1, p2, ncol = 2)
```

```
# Fit a Random Forest model on the training set with 100 ntrees
```

```
rf_model100 <- randomForest(Windspeed ~ time, data = train_data, ntree = 100)
# Display the Random Forest model summary
summary(rf_model100)

# Fit a Random Forest model on the training set with 200 ntrees
rf_model200 <- randomForest(Windspeed ~ time, data = train_data, ntree = 200)
# Display the Random Forest model summary
summary(rf_model200)
# Fit a Random Forest model on the training set with 500 ntrees
rf_model500 <- randomForest(Windspeed ~ time, data = train_data, ntree = 500)
# Display the Random Forest model summary
summary(rf_model500)

# Predict Windspeed values for the test set using the Random Forest model
rf_predictions100 <- predict(rf_model100, newdata = test_data)

# Calculate the root mean squared error (RMSE) for the Random Forest model
rf_rmse100 <- sqrt(mean((test_data$Windspeed - rf_predictions100)^2))
cat("Random Forest with 100 n trees RMSE:", rf_rmse100)

# Predict Windspeed values for the test set using the Random Forest model
rf_predictions200 <- predict(rf_model200, newdata = test_data)

# Calculate the root mean squared error (RMSE) for the Random Forest model
rf_rmse200 <- sqrt(mean((test_data$Windspeed - rf_predictions200)^2))
cat("Random Forest with 200 n trees RMSE:", rf_rmse200)

# Predict Windspeed values for the test set using the Random Forest model
```

```
rf_predictions500 <- predict(rf_model500, newdata = test_data)


# Calculate the root mean squared error (RMSE) for the Random Forest model

rf_rmse500 <- sqrt(mean((test_data$Windspeed - rf_predictions500)^2))

cat("Random Forest with 500 n trees RMSE:", rf_rmse500)



rf_rmse_together <- cbind(rf_rmse100, rf_rmse200, rf_rmse500)


barplot(rf_rmse_together, beside = TRUE)



#We can see that n trees 500 has lowest values, I will be using ntrees=500 as comparison with other machine learning,


# Compare the RMSE values of the Linear Regression, SVR, and Random Forest models

cat("Linear Regression RMSE:", rmse_LR)

cat("Radial SVR RMSE:", svr_rmse_radial)

cat("Random Forest with 500 ntrees RMSE:", rf_rmse500)


# Plot the actual vs. predicted values for the Linear Regression, SVR, and Random Forest models

p1 <- ggplot() +

  geom_point(data = test_data, aes(x = Windspeed, y = predictions), color = "blue") +

  geom_abline(slope = 1, intercept = 0, color = "red") +

  labs(title = "Linear Regression: Actual vs. Predicted Windspeed",

     x = "Actual Windspeed",

     y = "Predicted Windspeed") +

  theme_minimal()
```

```
p2 <- ggplot() +
  geom_point(data = test_data, aes(x = Windspeed, y = svr_predictions_radial), color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "SVR radial: Actual vs. Predicted Windspeed",
       x = "Actual Windspeed",
       y = "Predicted Windspeed") +
  theme_minimal()

p3 <- ggplot() +
  geom_point(data = test_data, aes(x = Windspeed, y = rf_predictions500), color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Random Forest: Actual vs. Predicted Windspeed",
       x = "Actual Windspeed",
       y = "Predicted Windspeed") +
  theme_minimal()

grid.arrange(p1, p2, p3, ncol = 3)

#ARIMA MODEL

AA101 = subset(AA99, select = c(Windspeed) )

AA101 <- as.data.frame(AA101)
```

PORTFOLIO ON PREDICTION/FORECASTING OF WINDSPEED AT MIDDLEBROUGH INTERNATIONAL AIRPORT USING TIME SERIES DATASET AND R PROGRAMMING LANGUAGE

```
set.seed(123)

train_indices <- sample(1:nrow(AA101), 0.8 * nrow(AA101))

train_data <- AA101[train_indices, ]

test_data <- AA101[-train_indices, ]

str(train_data)

str(test_data)



arima_model <- auto.arima(train_data, seasonal=TRUE, stepwise = TRUE)

arima_forecast <- forecast(arima_model, h=length(test_data))

print(arima_model)


arima_rmse <- sqrt(mean((test_data - arima_forecast$mean)^2))

arima_rmse


#Barploting all models RMSE together


all_rmse_together <- cbind(svr_rmse_radial, arima_rmse, rmse_LR, rf_rmse500 )


barplot(all_rmse_together, beside = TRUE)


#Random forest proves to be the best model.
```