Final Project

Priyanka Kaushal (24200862)

```
## code chunk to hide warnings
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

Part 1 : Analysis

Data Selected - Public bike sharing data from link below

https://datasetsearch.research.google.com/search?src=3&query=bike%20sharing&docid=L2cvMTFqY2ttcXNnZw%3D%3D

This dataset only contains collected data related to public bike sharing systems around the world. All the data are collected from open databases and website of the different public bike sharing operators. data is dated **2nd September 2019.**

Load and explain data : -

```
library(readxl)
ds <- read_excel("Bike_sharing.xlsx",sheet = "Data_for_cluster"
                    ,col_names = TRUE)
```

Size and structure of data set - the size (number of rows and columns)

Selected Data is having 75 columns and 64 rows.

```
library(tibble)
#str(ds, width = 60, strict.width = "wrap")
p = as.tibble(ds)
cat("Number of columns in dataframe:",ncol(p),"\n")
```

```
Number of columns in dataframe: 75
```

```
cat("Number of rows in dataframe:",nrow(p),"\n")
```

```
Number of rows in dataframe: 64
```

Remove and rename columns for further use

After removing columns now the dataset consists of 13 columns & 64 rows.

Check the datatypes of each selected columns

```
## rename first column name
colnames(p)[1] = "Serial"

## remove extra columns
p = subset(p , select = c(`Name`,   `Continent`,`Country`,`City`,`Climate`
                          ,`Operation`, `Population`,`Income`,`Owner`
                          ,`Operator`,  `Station`,`Docks`,`Bikes`))


cat("Number of columns in dataframe:",ncol(p),"\n")
```

```
Number of columns in dataframe: 13
```

```
cat("Number of rows in dataframe:",nrow(p),"\n")
```

```
Number of rows in dataframe: 64
```

```
knitr::kable(sapply(p,class))
```

	x
Name	character
Continent	character
Country	character
City	character
Climate	character
Operation	numeric
Population	numeric
Income	numeric

	x
Owner	character
Operator	character
Station	numeric
Docks	numeric
Bikes	numeric

Check for missing values and remove NA

Before doing any analysis we need to check and remove if there is any missing value.

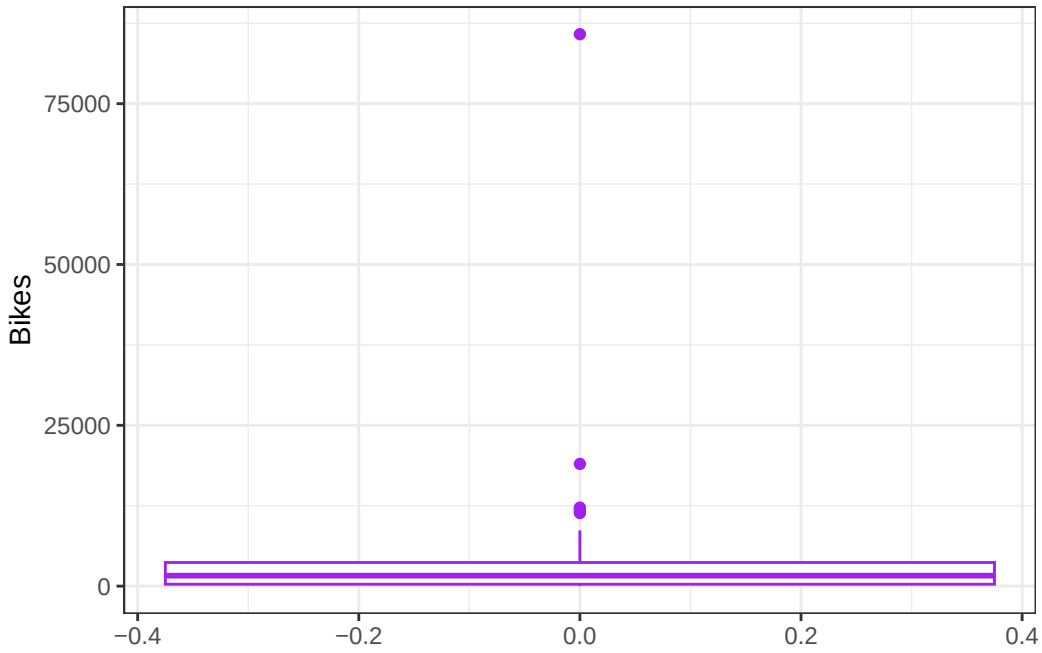
```
## exclude rows where there is NA for any columns
p= na.exclude(p)
```

Check the data for outliers .

There is one outlier in data which we should investigate and decide , it is wrong data and can be removed or its a valid data

- We found a new company was established with high number of stations and Docks in China which is shown as outlier .
- We can not remove this , as its valid and very important point to show in our analysis.

```
library(ggplot2)
ggplot(p, aes( y = Bikes))+geom_boxplot(col = 'purple')+
 labs('Boxplot of the number of bikes') + theme_bw()
```



Factorize categorical columns

1. Continent
2. Owner

```
Continent = factor(p$Continent,levels = c("Asia", "Australia", "America",
                                          "Europe"),ordered = TRUE)

Owner = factor(p$Owner,labels = c("Public","Private"),ordered = TRUE)
print(levels(Continent))
```

```
[1] "Asia"      "Australia" "America"   "Europe"
```

```
print(levels(Owner))
```

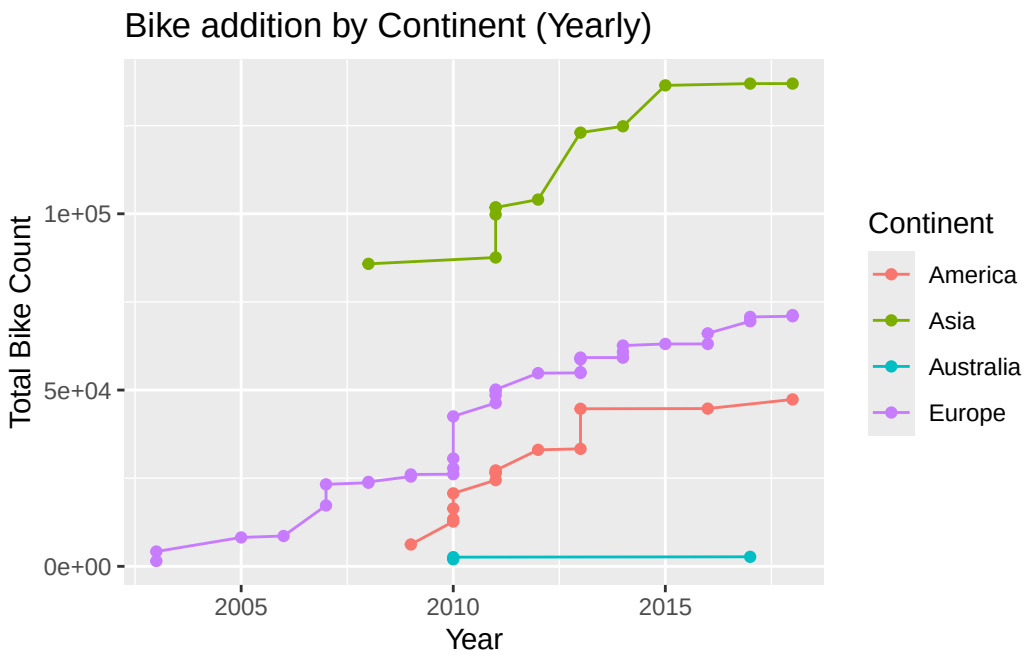
```
[1] "Public"  "Private"
```

Plot the increase of bike rentals added by companies established in a series of time per continent

Here in this graph we found below observations :

- Europe was the first continent where bike sharing companies were established before 2005.
- Slowly near 2010 other continents introduced this service .
- Asia has the highest number of bike added on the year 2008 when the system was introduced in this continent .
- Every continent is constantly increasing the bike service , but we see its constant in Australia.

```
library(ggplot2)
library(dplyr)

# mutate data to add a column showing cumutlative sum
p <- p %>%
  arrange(Operation) %>%  # sort data by the year
  group_by(Continent) %>% # separate cumulative sums for each continent
  mutate(CumulativeBikes = cumsum(Bikes))

#plot the cumusulative bike sum over year by continent
ggplot(p, aes(x = Operation , y = CumulativeBikes, color = Continent)) +
  geom_line() +
  geom_point() +
  labs(title = "Bike addition by Continent (Yearly)",
       x = "Year", y = "Total Bike Count")
```



Will see the summary of how many companies are established till data snapshot per country / continent

- USA followed by China & France has the highest number of bike sharing companies established .
- 19 total countries including South Korea,Mexico, Israel,India ,Brazil & Kazakhstan have only 1 company established .

```
library(dplyr)
# Summarize the number of unique bike companies by Continent and City
summary <- p %>%
  group_by(Continent, Country) %>%
  summarise(NumCompanies = n_distinct(Name), .groups = "drop") %>%
  arrange(desc(NumCompanies))


# View the summarized data
#print(summary)
knitr::kable(summary)
```

Continent	Country	NumCompanies
America	USA	10
Asia	China	5
Europe	France	4
Australia	Australia	3
Europe	Germany	3
Europe	Italy	3
Europe	Spain	3
America	Canada	2
Asia	Japan	2
Europe	Austria	2
Europe	Denmark	2
Europe	Greece	2
Europe	Poland	2
Europe	Switzerland	2
America	Brasil	1
America	Mexico	1
Asia	India	1
Asia	Israel	1
Asia	Kazakhstan	1
Asia	South Korea	1
Europe	Belgium	1
Europe	Croatia	1
Europe	Czech Republic	1
Europe	Finland	1
Europe	Hungary	1
Europe	Iceland	1
Europe	Ireland	1
Europe	Lithuania	1
Europe	Luxembourg	1
Europe	Norway	1
Europe	Portugal	1
Europe	Slovenia	1
Europe	UK	1

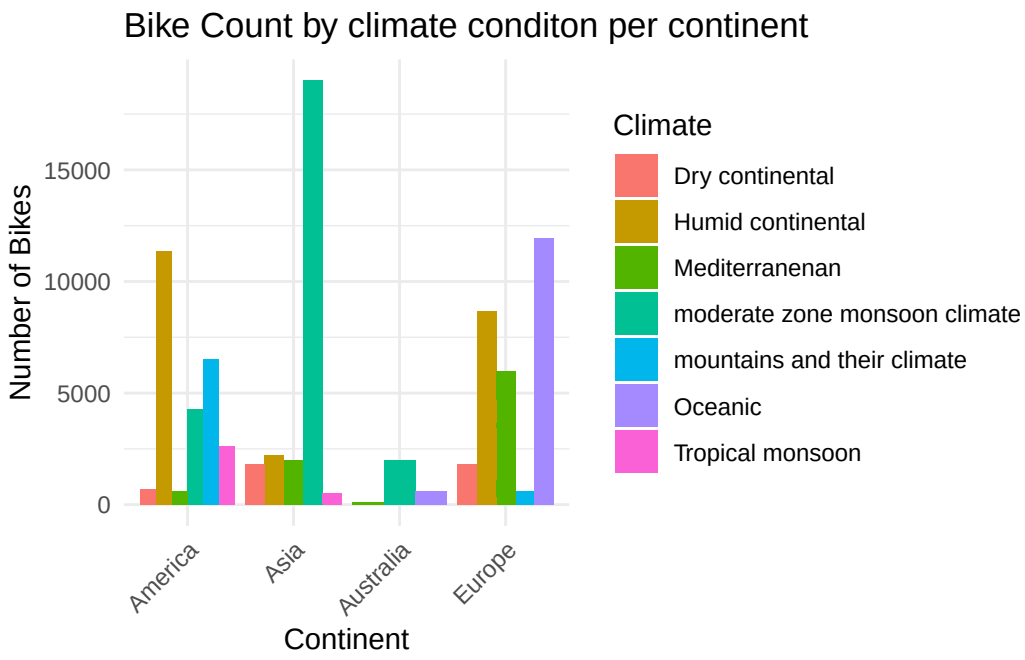
We will check if there is any impact of climate for Bike rental count .

There is no significant impact of climate in bike count as per the below graph .

I have plotted the graph by removing max bikes count as it was making all other points in the graph hard to read .

We don't see any particular climate standing out significantly , every continent has different climate which has highest bikes .

```
p_n <- p[p$Bikes != max(p$Bikes), ]

ggplot(p_n, aes(x = Continent , y = Bikes, fill = Climate)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Bike Count by climate conditon per continent ",
       x = "Continent", y = "Number of Bikes") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

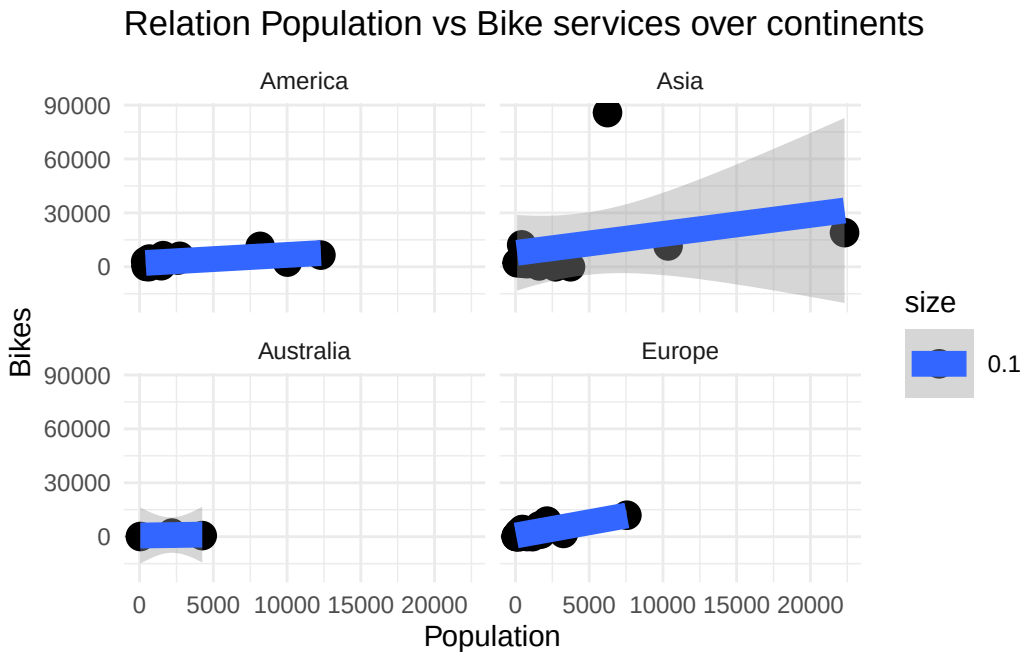


Check if there is any relation between population and number of bikes rented .

Using lm method i can see there is a linear relationship between bike count and population across all continents captured.

```
## linear model plot to see linear relationship

c <- ggplot(p,aes(x= Population,y=Bikes,size=0.1)) + geom_point()+
  stat_smooth(method=lm)+facet_wrap(~ Continent)+
  labs(title = "Relation Population vs Bike services over continents") +
```

```
  theme_minimal()

c
```



Summary

After exploring the dataset and analysis above, below are the observations found :

Europe was the first one where Bike data sharing company was established in 2004, after few years it started across the globe near 2009.

There is no impact of the climate or service provider on the bike company establishment , but we see a significant relation between population and establishment .

Conclusion

The analysis concludes that the growth and expansion of bike-sharing services across the globe were primarily driven by population rather than environmental factors or the nature of service providers.

Part 2: R Package (fable)

Dataset selected

Dataset used :- https://data.gov.ie/dataset/pfsa02-fertiliser-sales

This dataset contains information about fertiliser sales over past years , i will be using 2000 to 2019 year data.

Citation for selected package

I wanted to work on prediction of time series data so selected **fable -** this package is used for **forecasting** time series data. It builds on the **tsibble** package

```
citation("fable")
```

```
To cite package 'fable' in publications use:

  O'Hara-Wild M, Hyndman R, Wang E (2024). _fable: Forecasting Models
  for Tidy Time Series_. R package version 0.4.1,
  <https://CRAN.R-project.org/package=fable>.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {fable: Forecasting Models for Tidy Time Series},
    author = {Mitchell O'Hara-Wild and Rob Hyndman and Earo Wang},
    year = {2024},
    note = {R package version 0.4.1},
    url = {https://CRAN.R-project.org/package=fable},
  }
```

Import packages

```
library(readxl)
library(fable)
library(tsibble)
library(ggplot2)
```

Load and shape data for Prediction

Data must be shaped as time series data to be used in prediction model for forecasting .

```
# Load excel data
ed <- read_excel("FertiliserSales.xlsx",col_names = TRUE)

# select required column for analysis
ed  <- ed %>%
  select(Year,State,UNIT,VALUE) %>%
  filter( UNIT == "Tonnes",State == "State")

#select data from 2000 to 2019
ed <- ed %>%
  select(Year,VALUE) %>%
  filter(Year >= 2000 & Year <= 2019)

# Convert data into time series format
ed <- ed %>%
  as_tsibble(index = Year)
```

Functions used from package

- **model**() - package provides built -in model to use on your data for predictions , i am using ARIMA model for predicting future 5 years data based on 19years of data.
- **forecast**() - this generates forecast based on the fitted model
- **report**() - this gives report summary on the model performance.

function used to select model() ,then forecast() based on model selected and report() to view model

```
# put the data in predictor model
model <- ed %>%
  model(ARIMA(VALUE ~ pdq(1, 0, 0)))

# Forecast the next 5 years
forecast_values <- model %>%
  forecast(h = 5)

## report for the model
model %>% report()
```

```
Series: VALUE
Model: ARIMA(1,0,0) w/ mean

Coefficients:
         ar1   constant
      0.7323  397887.26
s.e.  0.1581   23176.83

sigma^2 estimated as 1.379e+10:  log likelihood=-261.18
AIC=528.36   AICc=529.86   BIC=531.35
```

```r
## put the predicted value in datframe
fv <- data.frame(forecast_values)

## filter to select only year & value
fv <- fv %>%
  select(Year,VALUE=.mean)


# stack rows from observation & model prediction
final <- bind_rows(ed,fv)

# divide the final dataframe into 2 parts to be used in plot
before_2019 <- final %>% filter( Year <= 2019)
after_2019 <- final %>% filter(Year >= 2019)

#convert Value to numeric for plotting
final$VALUE <- as.numeric(final$VALUE)

cat("***Predicted Value for future Years :***")
```

```
***Predicted Value for future Years :***
```

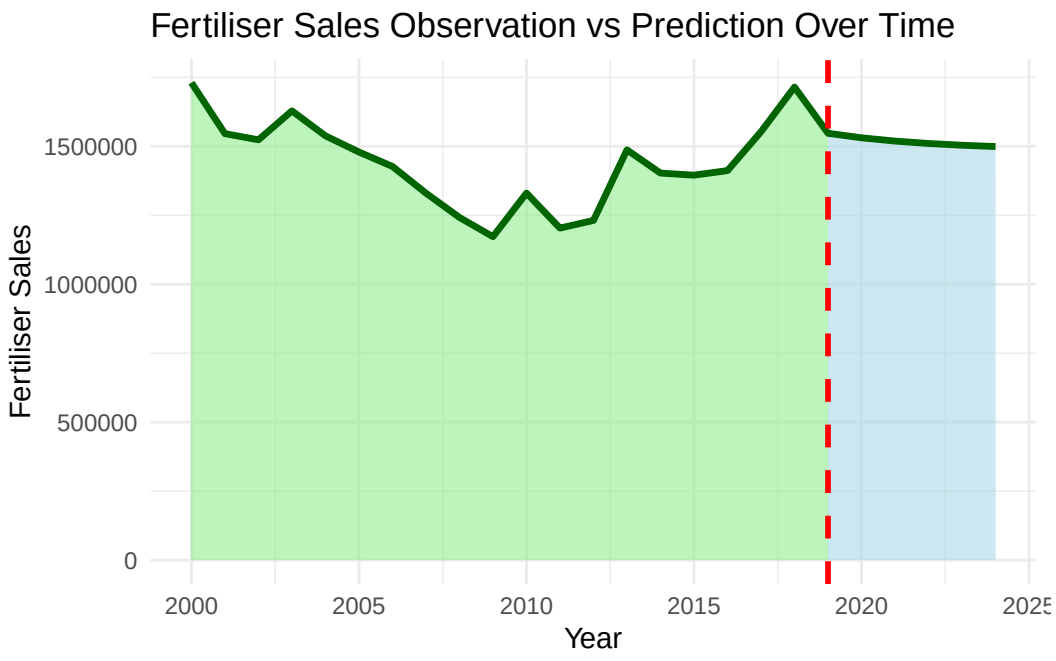
```r
print(fv)
```

```
  Year   VALUE
1 2020 1530834
2 2021 1518936
3 2022 1510222
4 2023 1503841
5 2024 1499168
```

Plot to see Observation vs Predicted data

Using the predicted value from model alongwith the observation data fed in the model , i am using ggplot area plot to show the observation vs predicted data for fertilizer YoY sales.

```
## Area Plot to see Observation vs Predicted data
ggplot() +
  # Area plot for data before 2019
  geom_area(data = before_2019, aes(x = Year, y = VALUE),
            fill = "lightgreen", alpha = 0.6) +
  # Area plot for data after 2019 with a different color
  geom_area(data = after_2019, aes(x = Year, y = VALUE),
            fill = "lightblue", alpha = 0.6) +
  # Line plot for the entire data
  geom_line(data = final, aes(x = Year, y = VALUE),
            color = "darkgreen", size = 1.2) +
  # Add a vertical line at year 2019
  geom_vline(xintercept = 2019, linetype = "dashed",
             color = "red", size = 1) +
  labs(title = "Fertiliser Sales Observation vs Prediction Over Time"
       , x = "Year", y = "Fertiliser Sales") +
  theme_minimal()
```



Part 3 : Functions/Programming

Define function

I am defining f-test function which takes input of dataset and 2 variables to compare its variance , this can be used to decide that variance are significantly different or not , which is concluded using **p-value** if its high (greater than 0.05), we fail to reject the null hypothesis and conclude that the variances are not significantly different.

Explanation:

F-test:-This function works to perform f-test on two variables in a dataset , get key statistics values from a numerical column in R, store the results in an S3 class, and define methods for printing, summarizing, and plotting the results. It takes input argument of a dataframe , 2 variables in the dataframe

- checks if dataset is valid , and have variables as passed from the function input argument.
- compares the variance between 2 variables , provides F-statistic , p-value & confidence interval.
- Also summarises the key statistics like the mean, median, standard deviation, and the 95% confidence interval for a 2 variables .
- The result is stored as an S3 class **ftest**

```
# Define the S3 class and function
# F-test function
f_test <- function(data, var1, var2) {

  # Ensure 'data' is a data frame
  if (!is.data.frame(data)) {
    stop("Input data must be a data frame")
  }

  # Ensure var1 and var2 are columns in the data
  if (!(var1 %in% names(data)) | !(var2 %in% names(data))) {
    stop("The specified columns are not present in the dataset")
  }

  # check if data is numeric
```

```r
if (!(is.numeric(data[[var1]])) | !(is.numeric(data[[var2]])))
{
  stop("Data must be a numeric vector.")
}

# Perform basic statistical analysis for both variable
## var 1 numerical analysis
mean_val <- mean(data[[var1]])
median_val <- median(data[[var1]])
sd_val <- sd(data[[var1]])
n <- length(data[[var1]])

## var 2 numerical analysis
mean_val1 <- mean(data[[var2]])
median_val1 <- median(data[[var2]])
sd_val1 <- sd(data[[var2]])
n1 <- length(data[[var2]])

# Calculate the 95% Confidence Interval for the mean
##var1
error_margin <- qt(0.975, df = n - 1) * sd_val / sqrt(n)
conf_interval <- c(mean_val - error_margin, mean_val + error_margin)

## var2
error_margin1 <- qt(0.975, df = n1 - 1) * sd_val1 / sqrt(n1)
conf_interval1 <- c(mean_val1 - error_margin1, mean_val1 + error_margin1)


# Perform F-test (comparing variances)
f_test_result <- var.test(data[[var1]], data[[var2]])

# Create an S3 class for the result
result <- list(
  statistic = f_test_result$statistic,
  p_value = f_test_result$p.value,
  var1 = var1,
  var2 = var2,
  conf_int = f_test_result$conf.int,
  mean_val_var1= mean_val,
  mean_val_var2 = mean_val1,
  median_val_var1 = median_val,
  median_val_var2 = median_val1,
```

```
    sd_val_var1 = sd_val,
    sd_val_var2 = sd_val1,
    error_margin_var1 = error_margin,
    error_margin_var2 = error_margin1,
    conf_interval_var1 = conf_interval,
    conf_interval_var2 = conf_interval1
  )

  # Assign the class "ftest"
  class(result) <- "ftest"

  return(result)
}
```

Create print method

print() - provides a overview on the f-test result by showing

- p-value
- F-statistic
- Confidence Interval

```
# Print method for ftest class
print.ftest <- function(x) {
  cat("F-test result:\n")
  cat("Comparing variances of", x$var1, "and", x$var2, "\n")
  cat("F-statistic:", x$statistic, "\n")
  cat("p-value:", x$p_value, "\n")
  cat("95% Confidence interval for the ratio of variances:", "[",
      x$conf_int, "]","\n\n")
}
```

Create summary method

summary() - Provides a statistical report on the analysis of 2 variables

- mean
- median
- standard deviation

- Error margin
- confidence interval

```
# Summary method for ftest class
summary.ftest <- function(x) {
  cat("Numerical Summary of selected variables:\n", x$var1, "and",
      x$var2,"\n")
  cat("**********************************:\n")
  cat("Mean values:", x$mean_val_var1,",",x$mean_val_var2,"\n")
  cat("Median values:", x$median_val_var1,",",x$median_val_var2,"\n")
  cat("Standard Deviation values:", x$sd_val_var1,",",x$mean_val_var2,"\n")
  cat("Error Margin values:", x$error_margin_var1,",",x$median_val_var2,"\n")
  cat("Confidence Interval values:","[", x$conf_interval_var1,"]",","
      ,"[",x$conf_interval_var2,"]","\n")

}
```

Create plot method

plot() - this plots the F-test confidence interval for ratio of variance among two variables alongwith the mean of confidence interval upper & lower bound.

```
# Plot method for ftest class
plot.ftest <- function(x) {
  # plot of the confidence interval
  plot(c(0, 6), c(x$conf_int[1]-1, x$conf_int[2]+1), type = "n",
       xlab = "Confidence Interval", ylab = "", xaxt = "n")
  # Plot the confidence interval line
  segments(x$conf_int[1], 3, x$conf_int[2], 3, col = "blue", lwd = 2)
  # Mark the mean of the confidence interval with a red point
  points(mean(x$conf_int), 3, col = "red", pch = 16, cex = 2)
  # Add x-axis with the confidence interval range
  axis(1, at = seq(x$conf_int[1], x$conf_int[2], length.out = 5),
      labels = paste0(round(seq(x$conf_int[1], x$conf_int[2],
                                length.out = 5), 3)))
  # Title
  title(main = paste("F-test C.I for","[", x$var1,
                     "and", x$var2,"]"))
}
```

Register S3 function with class

- Register the S3 methods defined for a specific function and class combination.
- this ensures that when the generic function (e.g summary, print, plot) is called on an object of class ftest , the appropriate method (in this case, summary.ftest,print.ftest,plot.ftest) is dispatched respectively.

```
# register the method - only needed if working with Quarto or RMarkdown.
registerS3method("print", "ftest", print.ftest)
registerS3method("summary", "ftest", summary.ftest)
registerS3method("plot", "ftest", plot.ftest)
isS3method("print.ftest")
```

```
[1] TRUE
```

```
isS3method("summary.ftest")
```

```
[1] TRUE
```

```
isS3method("plot.ftest")
```

```
[1] TRUE
```

Working example

I am using **iris** dataset to check the **ftest** function created and variables i will input to test are **sepal length & width**

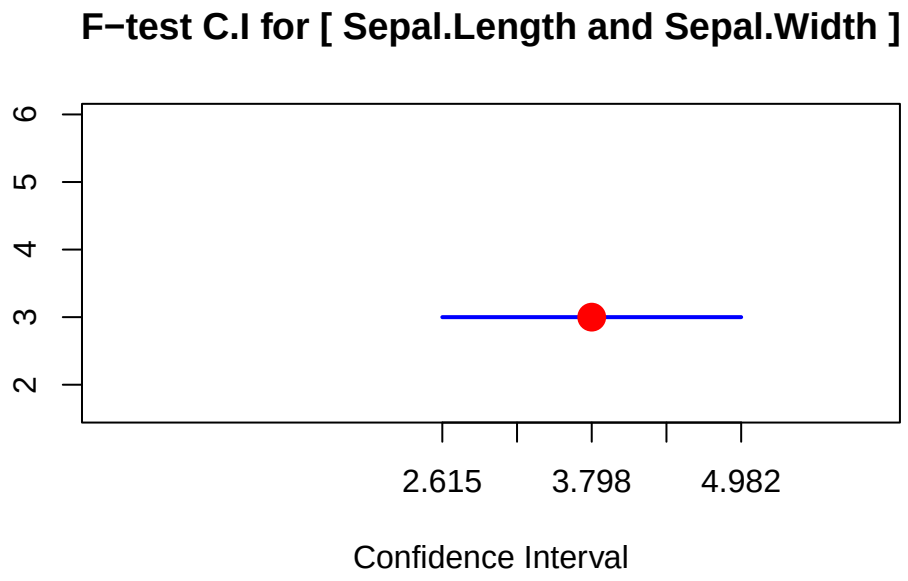
```
# Example Usage of the f_test function with iris dataset

# Load iris dataset
data(iris)

# Perform F-test comparing variances of Sepal.Length and Sepal.Width
ftest_result <- f_test(iris, "Sepal.Length", "Sepal.Width")

# Display the result using print()
print(ftest_result)
```

```
F-test result:
Comparing variances of Sepal.Length and Sepal.Width
F-statistic: 3.609304
p-value: 3.597123e-14
95% Confidence interval for the ratio of variances: [ 2.614702 4.982242 ]
```

```
# Display the summary of the result
summary(ftest_result)
```

```
Numerical Summary of selected variables:
 Sepal.Length and Sepal.Width
**********************************:
Mean values: 5.843333 , 3.057333
Median values: 5.8 , 3
Standard Deviation values: 0.8280661 , 3.057333
Error Margin values: 0.1336009 , 3
Confidence Interval values: [ 5.709732 5.976934 ] , [ 2.98701 3.127656 ]
```

```
# Display the plot of the result
plot(ftest_result)
```



The function output explains the variability between two variables are significantly different, as **p-value** ~ **3.6*10^-14** its very low (less than 0.05), we reject the null hypothesis and

conclude that the variances are significantly different. It also explains that Sepal Length mean is 5.84 , median is 5.8 , std deviation = 0.82 . For Sepal Width mean is 3.06, median is 3 , std deviation = 3.06.