```
---
title: 'Week 1: Intro to R'
output: html_document
date: "2023-05-31"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

# Welcome to RStudio Intro Tutorial

# How to import files for data
setwd("/Users/angelrae301/Desktop/INBRE Bioinformatics/Week 1") #This is where
your files are stored, so you are telling R to only work witin this folder
df <- read.csv("HW1_sewingthread.csv")
head(df) #used to get the first parts of the table, looks at the first 6 lines
str(df) #structure, how it is sorted
head(df)
summary(df)

# Print a welcome message
print("Welcome to RStudio!")

# Basic arithmetic operations
x <- 10
y <- 5
sum <- x + y
diff <- x - y
product <- x * y
quotient <- x / y

# Print the results
print("Basic Arithmetic Operations:")
print(paste("Sum:", sum))
print(paste("Difference:", diff))
print(paste("Product:", product))
print(paste("Quotient:", quotient))

# Creating and manipulating vectors
vector1 <- c(1, 2, 3, 4, 5)
vector2 <- c(6, 7, 8, 9, 10)

# Print the vectors
print(vector1)
print(vector2)

# Vector operations
vector_sum <- vector1 + vector2
vector_diff <- vector1 - vector2
vector_product <- vector1 * vector2

# Basic data analysis
weight <- c(69, 62, 57, 59, 59, 64, 56, 66, 67, 66)
mean(weight)
median(weight)
max(weight)
min(weight)
```
```

```
var(weight) # to calculate the variance
sd(weight) #standard deviation
range(weight)

#There is another way to calculate summary statistics (mean, median, min and max)
height <- c(112, 102, 83, 84, 99, 90, 77, 112, 133, 112)
library(psych)
describe(height, type = 2)

#To extract specific portions or your vector
some_values <- height[c(2,3,9,10)] # This will extract the 2nd, 3rd, 9th, and 10th
values

height_sorted <- sort(height) # this will sort in the ascending order.

#Can also be a Fancy Calculator
log(12.43)
log10(12.43) #log to base 10
log(12.43, base = 10) #alternative way of writing the log10 function
sqrt(12.43)
exp(12.43)

#How can you ask for help
help(mean)
?mean # this can also be typed into the bottom portion of the RStudio

#Another important function is head(), this allows you to view the first few lines
of the dataframe.

#ANOVAs
#First is to make sure that you have the required packages installed and loaded

install.packages("tidyverse")
library(tidyverse)

#Ensure your data is in the appropriate format, with the dependent variable in one
column and the grouping variable in another column. for this example the data set
will be reffered to as 'data' with a dependent variable 'y' and a grouping
variable 'group'

#Fit ANOVA model
model <- lm(y ~ group, data = data)

#Perform ANOVA
anova_result <- anova(model)

#Print ANOVA table
print(anova_result)

#This ANOVA table will provide you with various statistics, including the sum of
squares, degrees of freedom, mean squares, and the F-statistic. The p-value
associated with the F-statistic indicates the significance of the group
differences. A p-value below a certain threshold (eg. 0.05) suggests that there
are significant differences between the groups/

#It is important to note that ANOVA assumes certain assumptions, such as normality
and homogeneity of variances.
```

#the tilde ~ is used to define the relationship between dependent variable and independent variables. The variable on the left-hand side of the tilde operator is the dependent variable and the variable(s) on the right hand side of tilde operator is/are called the independent variables
```

```{r}
# Welcome to Intro to ggplot2 Tutorial

# Install and load the ggplot2 package
install.packages("ggplot2")
library(ggplot2)

# Create a sample dataset
data <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(3, 5, 4, 6, 8)
)

# Scatter plot
ggplot(data, aes(x = x, y = y)) +
  geom_point()

# Line plot
ggplot(data, aes(x = x, y = y)) +
  geom_line()

# Bar plot
ggplot(data, aes(x = x, y = y)) +
  geom_bar(stat = "identity")

# Box plot
ggplot(data, aes(x = x, y = y)) +
  geom_boxplot()

# Histogram
ggplot(data, aes(x = y)) +
  geom_histogram()

# Customize plot appearance
ggplot(data, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Scatter Plot", x = "X", y = "Y") +
  theme_minimal()

#xlab() is another way to incorporate an axis title

# Faceted plot
data2 <- data.frame(
  group = c("A", "A", "B", "B"),
  value = c(3, 5, 4, 6))

ggplot(data2, aes(x = group, y = value)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ group)
```
```

```
# Save the plot as a PNG file
ggsave("plot.png")
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.