7.1.Create a 2x3x2 array with values from 1 to 12.

a. **Name the rows as "Row1", "Row2", and columns as "Col1", "Col2", "Col3".**

b. **Access the elements in the first row and second column of the second layer.**

c. **Replace all elements greater than 6 with NA.**

d. **Calculate the sum of all elements in the array.**

#Code:

```
# a: Create the array with names
my_array <- array(1:12, dim = c(2, 3, 2),
          dimnames = list(c("Row1", "Row2"),
                  c("Col1", "Col2", "Col3")))
print(my_array)

# b: Access element in first row, second column, second layer
element <- my_array["Row1", "Col2", 2]
print(element)

# c: Replace values > 6 with NA
my_array[my_array > 6] <- NA
print(my_array)

# d: Calculate sum ignoring NA
total_sum <- sum(my_array, na.rm = TRUE)
print(total_sum)
```

#output:

```
, , 1
   Col1 Col2 Col3
Row1    1    3    5
Row2    2    4    6
, , 2
   Col1 Col2 Col3
Row1    7    9   11
Row2    8   10   12
```

```
[1] 9

, , 1
   Col1 Col2 Col3
Row1    1    3    5
Row2    2    4    6
, , 2
   Col1 Col2 Col3
Row1   NA   NA   NA
Row2   NA   NA   NA

[1] 21
```

7.2.Create a factor using the vector c("Red", "Blue", "Green", "Red", "Blue").

a. **Display the levels of the factor.**
b. **Generate a factor with 3 levels ("High", "Medium", "Low") repeated 4 times.**
c. **Modify the factor to include a new level "Very High".**
d. **Drop the level "Medium" from the factor and display the updated factor.**

#Code:

```
# Step 1: Create a factor
colors <- factor(c("Red", "Blue", "Green", "Red", "Blue"))

# Step a: Display levels
print(levels(colors))

# Step b: Create repeated factor
levels_vector <- factor(rep(c("High", "Medium", "Low"), times = 4))
print(levels_vector)

# Step c: Add new level
levels(levels_vector) <- c(levels(levels_vector), "Very High")
```

```
# Step d: Drop "Medium" and display updated factor
levels_vector <- levels_vector[levels_vector != "Medium"]
levels_vector <- droplevels(levels_vector)
print(levels_vector)
```

#output:

```
[1] "Blue"  "Green" "Red"

[1] High    Medium Low    High    Medium Low    High    Medium Low    High    Medium Low
Levels: High Low Medium

[1] High Low   High Low   High Low   High Low
Levels: High Low
```

8. Write a script to process the text "R Programming is Fun and Challenging".

a. Extract every second word from the sentence.

b. Count the number of occurrences of vowels (a, e, i, o, u) in the string.

c. Replace the word "Challenging" with "Exciting".

Code:

```
# Load required library
library(stringr)

# Original text
text <- "R Programming is Fun and Challenging"

# a. Extract every second word
words <- strsplit(text, " ")[[1]]
second_words <- words[seq(2, length(words), by = 2)]
print(second_words)

# b. Count number of vowels
```

```
lower_text <- tolower(text)
vowel_count <- sum(str_count(lower_text, "[aeiou]"))
print(vowel_count)

# c. Replace "Challenging" with "Exciting"
new_text <- sub("Challenging", "Exciting", text)
print(new_text)
```

#Output:

```
[1] "Programming"     "Fun"     "Challenging"

[1] 9

[1] "R Programming is Fun and Exciting"
```

9.1. Create a nested list containing:

a. **A data frame with student names, marks, and grades.**
b. **A vector with the total marks for each student.**
c. **A list of factors indicating the performance category ("Excellent", "Good", "Average").**

Code:

```
# a. Data frame with student names, marks, and grades
students_df <- data.frame(
  Name = c("Alice", "Bob", "Charlie"),
  Marks = c(88, 75, 62),
  Grade = c("A", "B", "C")
)

# b. Vector with total marks (assuming one subject each for now)
total_marks <- c(88, 75, 62)
```

```
# c. List of factors for performance category
performance <- list(
  factor(c("Excellent", "Good", "Average"))
)

# Combine all into a nested list
nested_list <- list(
  StudentData = students_df,
  TotalMarks = total_marks,
  PerformanceCategory = performance
)

# View the nested list
print(nested_list)

#output:
$StudentData
     Name Marks Grade
1   Alice    88     A
2     Bob    75     B
3 Charlie    62     C

$TotalMarks
[1] 88 75 62

$PerformanceCategory
$PerformanceCategory[[1]]
[1] Excellent Good      Average
Levels: Average Excellent Good
```

9.2. Write a script to:

a. **Access and modify the data frame inside the nested list.**

b. **Add a new entry for a student.**

c. **Extract students with "Excellent" performance.**

Code:

```
# Step a: Modify Bob's marks
nested_list$StudentData$Marks[nested_list$StudentData$Name == "Bob"] <- 80

# Step b: Add new student David
new_student <- data.frame(Name = "David", Marks = 91, Grade = "A")
nested_list$StudentData <- rbind(nested_list$StudentData, new_student)
nested_list$TotalMarks <- c(nested_list$TotalMarks, 91)
nested_list$PerformanceCategory[[1]] <- factor(
  c(as.character(nested_list$PerformanceCategory[[1]]), "Excellent"),
  levels = c("Average", "Good", "Excellent")
)

# Step c: Extract students with "Excellent" performance (Grade A)
excellent_students <- nested_list$StudentData[nested_list$StudentData$Grade == "A", ]
print(excellent_students)
```

Output:

```
   Name   Marks  Grade
1  Alice  88      A
4  David  91      A
```

Question 10:

Code:

```
# Install missing package if needed
if (!require("corrplot")) install.packages("corrplot")

# Load libraries
library(dplyr)
library(ggplot2)
library(readr)
library(lubridate)
library(corrplot)

# Load dataset
avocado <- read_csv("C:/Users/Suman Bajani/OneDrive/Desktop/avocado.csv")

# View structure and dimensions
str(avocado)
cat("Number of rows:", nrow(avocado), "\n")
cat("Number of columns:", ncol(avocado), "\n")

# Check for duplicates and missing values
duplicates <- avocado[duplicated(avocado), ]
cat("Number of duplicate rows:", nrow(duplicates), "\n")
print(colSums(is.na(avocado)))
missing_summary <- sapply(avocado, function(x) sum(is.na(x)))
cat("Total missing values in dataset:", sum(is.na(avocado)), "\n")



# Clean missing values
avocado_clean <- na.omit(avocado)
```

```r
cat("Rows after removing missing values:", nrow(avocado_clean), "\n")

# Summary statistics
mean_price <- mean(avocado_clean$AveragePrice)
median_price <- median(avocado_clean$AveragePrice)
sd_price <- sd(avocado_clean$AveragePrice)
cat("Mean:", mean_price, "\n")
cat("Median:", median_price, "\n")
cat("Standard Deviation:", sd_price, "\n")

# Group by type
avg_price_by_type <- avocado_clean %>%
  group_by(type) %>%
  summarise(AveragePrice = mean(AveragePrice))
print(avg_price_by_type)

# Standardize name (only if needed)
names(avocado_clean)[names(avocado_clean) == "Total Volume"] <- "Total.Volume"

# Scatter plot
# Scatter plot to visualize Total.Volume vs AveragePrice
plot <- ggplot(avocado_clean, aes(x = Total.Volume, y = AveragePrice)) +
  geom_point(alpha = 0.5, color = "darkgreen") +
  labs(title = "Total Volume vs Average Price",
       x = "Total Volume", y = "Average Price")
print(plot)

# Boxplot
ggplot(avocado_clean, aes(x = type, y = AveragePrice, fill = type)) +
  geom_boxplot() +
  labs(title = "Average Price by Avocado Type", x = "Type", y = "Average Price")
```

```
# Date conversion and line plot
avocado_clean$Date <- as.Date(avocado_clean$Date)
ggplot(avocado_clean, aes(x = Date, y = AveragePrice, color = type)) +
  geom_line() +
  labs(title = "Average Price Over Time by Avocado Type", x = "Date", y = "Average Price")

# Top 10 regions
top_regions <- avocado_clean %>%
  group_by(region) %>%
  summarise(AvgPrice = mean(AveragePrice)) %>%
  arrange(desc(AvgPrice)) %>%
  slice(1:10)
ggplot(top_regions, aes(x = reorder(region, AvgPrice), y = AvgPrice)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Regions by Average Price", x = "Region", y = "Average Price")

#Correlation
numeric_data <- select(avocado_clean, where(is.numeric))
cor_matrix <- cor(numeric_data, use = "complete.obs")
print(cor_matrix)
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.cex = 0.8,
         title = "Correlation Matrix", mar = c(0, 0, 1, 0))
```

#output:

Number of rows: 18249

Number of columns: 14

Number of duplicate rows: 0

Total missing values in dataset: 0

Rows after removing missing values: 18249

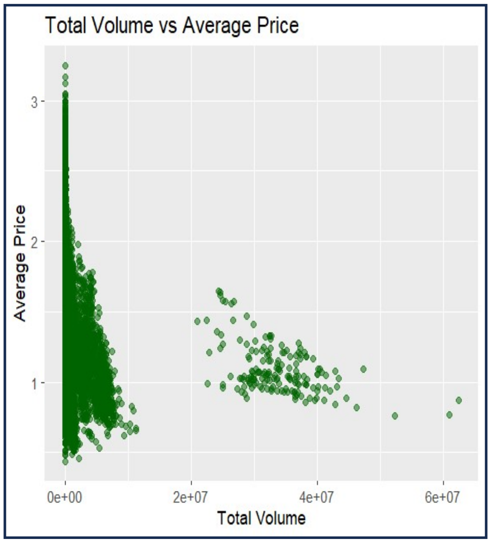
Mean: 1.405978

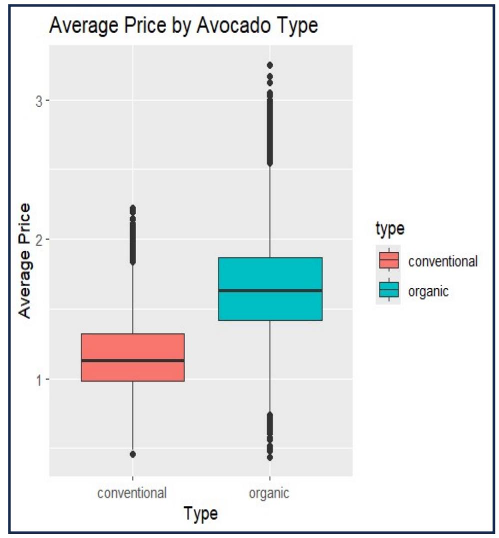
Median: 1.37

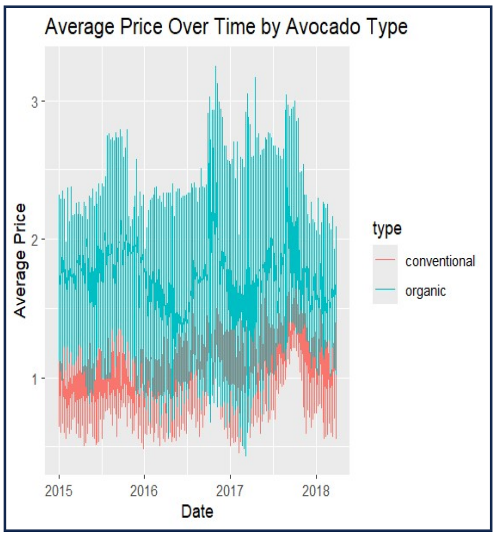
Standard Deviation: 0.4026766

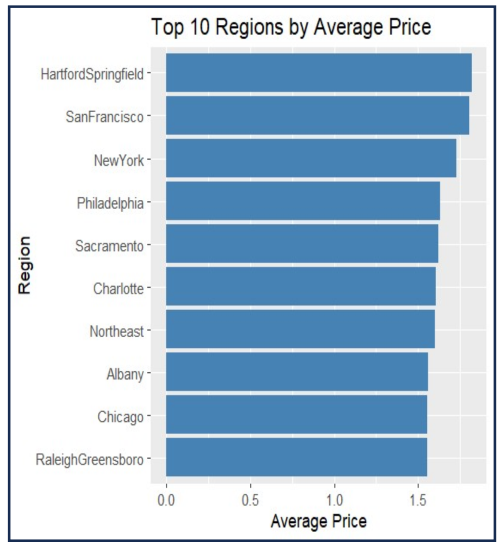
A tibble: 2 × 2

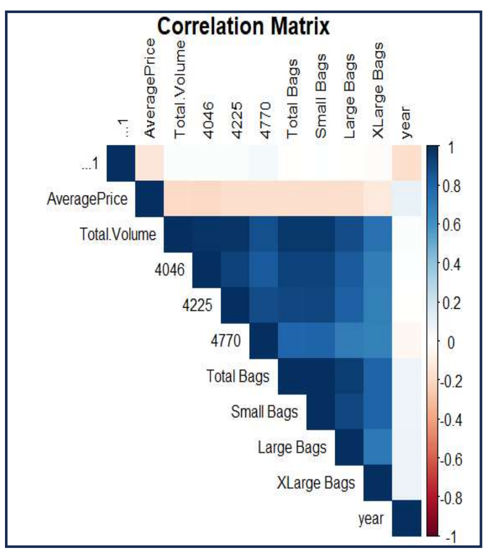
	type	AveragePrice
	<chr>	*<dbl>*
1	conventional	1.16
2	organic	1.65


Total Volume vs Average Price
Average Price
3
2
1
0e+00
2e+07
4e+07
6e+07
Total Volume


Average Price by Avocado Type
Average Price
3
2
1
conventional
organic
Type
type
conventional
organic


Average Price Over Time by Avocado Type
Average Price
3
2
1
2015
2016
2017
2018
Date
type
conventional
organic


Top 10 Regions by Average Price
Region
HartfordSpringfield
SanFrancisco
NewYork
Philadelphia
Sacramento
Charlotte
Northeast
Albany
Chicago
RaleighGreensboro
0.0
0.5
1.0
1.5
Average Price


Correlation Matrix
...1
AveragePrice
Total.Volume
4046
4225
4770
Total Bags
Small Bags
Large Bags
XLarge Bags
year
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1