RWorksheet\_Lapso#4b.Rmd

Darlene Erl Lapso

2023-11-28

1. Using the for loop, create an R script that will display a 5x5 matrix as shown in Figure 1. It must contain vectorA = [1,2,3,4,5] and a 5 x 5 zero matrix. Hint Use abs() function to get the absolute value

vectorA <- c(1,2,3,4,5)  
matzero\_A <- matrix(0, nrow = 5, ncol = 5)  
  
mat\_A <- matzero\_A  
for(i in 1:5) {  
 for (j in 1:5) {  
 mat\_A[i, j] <- abs(vectorA[i] - vectorA[j])  
 }  
}  
  
print(mat\_A)

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0 1 2 3 4  
## [2,] 1 0 1 2 3  
## [3,] 2 1 0 1 2  
## [4,] 3 2 1 0 1  
## [5,] 4 3 2 1 0

1. Print the string “\*” using for() function. The output should be the same as shown in Figure

n <- c()  
 for(i in 1:5) {  
 for(j in 1:i+1) {  
 n = c(n, "\*")  
 }  
print(n)  
n <- c()  
}

## [1] "\*"  
## [1] "\*" "\*"  
## [1] "\*" "\*" "\*"  
## [1] "\*" "\*" "\*" "\*"  
## [1] "\*" "\*" "\*" "\*" "\*"

1. Get an input from the user to print the Fibonacci sequence starting from the 1st input up to 500. Use repeat and break statements. Write the R Scripts and its output.

n <- as.integer(readline(prompt = "Enter the number of terms: "))

## Enter the number of terms:

a <- 0  
b <- 1  
cat("Fibonacci Sequence:", a, b)

## Fibonacci Sequence: 0 1

repeat {  
c <- a + b  
if (c > 500) {  
break  
}  
cat(", ",c)  
a <- b  
b <- c  
}

## , 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377

1. Import the dataset as shown in Figure 1 you have created previously.

library(readxl)  
shoe <- read\_excel("SHOE.xlsx")

## New names:  
## • `Shoe Size` -> `Shoe Size...1`  
## • `Height` -> `Height...2`  
## • `Gender` -> `Gender...3`  
## • `Shoe Size` -> `Shoe Size...4`  
## • `Height` -> `Height...5`  
## • `Gender` -> `Gender...6`

1. What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset? Show your codes and its result

head(shoe, 6)

## # A tibble: 6 × 6  
## `Shoe Size...1` Height...2 Gender...3 `Shoe Size...4` Height...5 Gender...6  
## <dbl> <dbl> <chr> <dbl> <dbl> <chr>   
## 1 6.5 66 F 13 77 M   
## 2 9 68 F 11.5 7 M   
## 3 8.5 64.5 F 8.5 59 F   
## 4 8.5 65 F 5 62 F   
## 5 10.5 70 M 10 72 M   
## 6 7 64 F 6.5 66 F

maleSub <- subset(shoe, Gender...3 == "M")  
femaleSub <- subset(shoe, Gender...3 == "F")

1. Create a subset for gender(female and male). How many observations are there in Male? How about in Female? Write the R scripts and its output.

Male <- nrow(maleSub)  
Female <- nrow(femaleSub)  
  
cat("Number of observations of Male in Gender...3 is: ", Male, "\n")

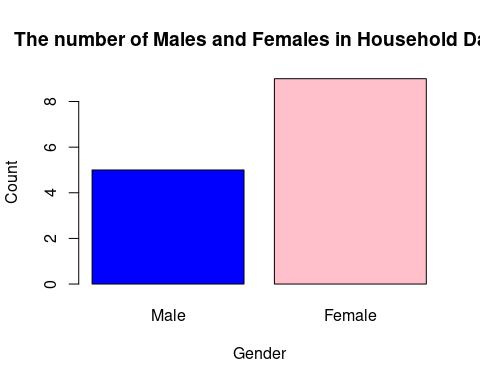
## Number of observations of Male in Gender...3 is: 5

cat("Number of observations of Female in Gender...3 is: ", Female, "\n")

## Number of observations of Female in Gender...3 is: 9

1. Create a graph for the number of males and females for Household Data. Use plot(), chart type = barplot. Make sure to place title, legends, and colors. Write the R scripts and its result.

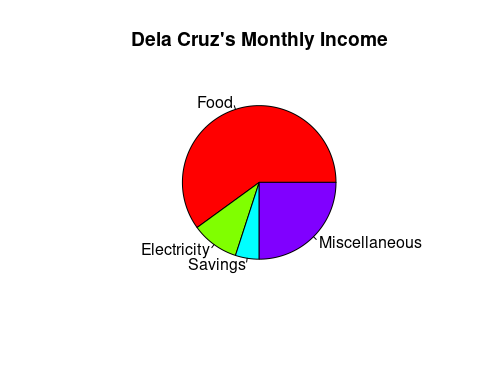
count <- c(Male, Female)  
  
gender <- c("Male", "Female")  
barplot(count,  
 names.arg = gender,  
 main = "The number of Males and Females in Household Data",  
 xlab = "Gender",  
 ylab = "Count",  
 col = c("blue", "pink"))



#legend("topright",  
# legend = gender,  
# fill = c("blue", "pink"))

* 1. Create a piechart that will include labels in percentage.Add some colors and title of the chart. Write the R scripts and show its output.

DelaCruz <- c(60, 10, 5, 25)  
 pie(DelaCruz,   
 main = "Dela Cruz's Monthly Income",  
 col = rainbow(length(DelaCruz)),   
 labels = c("Food", "Electricity", "Savings", "Miscellaneous"))



1. Use the iris dataset. data(iris)
2. Check for the structure of the dataset using the str() function. Describe what you have seen in the output.

data(iris)  
iris

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa  
## 7 4.6 3.4 1.4 0.3 setosa  
## 8 5.0 3.4 1.5 0.2 setosa  
## 9 4.4 2.9 1.4 0.2 setosa  
## 10 4.9 3.1 1.5 0.1 setosa  
## 11 5.4 3.7 1.5 0.2 setosa  
## 12 4.8 3.4 1.6 0.2 setosa  
## 13 4.8 3.0 1.4 0.1 setosa  
## 14 4.3 3.0 1.1 0.1 setosa  
## 15 5.8 4.0 1.2 0.2 setosa  
## 16 5.7 4.4 1.5 0.4 setosa  
## 17 5.4 3.9 1.3 0.4 setosa  
## 18 5.1 3.5 1.4 0.3 setosa  
## 19 5.7 3.8 1.7 0.3 setosa  
## 20 5.1 3.8 1.5 0.3 setosa  
## 21 5.4 3.4 1.7 0.2 setosa  
## 22 5.1 3.7 1.5 0.4 setosa  
## 23 4.6 3.6 1.0 0.2 setosa  
## 24 5.1 3.3 1.7 0.5 setosa  
## 25 4.8 3.4 1.9 0.2 setosa  
## 26 5.0 3.0 1.6 0.2 setosa  
## 27 5.0 3.4 1.6 0.4 setosa  
## 28 5.2 3.5 1.5 0.2 setosa  
## 29 5.2 3.4 1.4 0.2 setosa  
## 30 4.7 3.2 1.6 0.2 setosa  
## 31 4.8 3.1 1.6 0.2 setosa  
## 32 5.4 3.4 1.5 0.4 setosa  
## 33 5.2 4.1 1.5 0.1 setosa  
## 34 5.5 4.2 1.4 0.2 setosa  
## 35 4.9 3.1 1.5 0.2 setosa  
## 36 5.0 3.2 1.2 0.2 setosa  
## 37 5.5 3.5 1.3 0.2 setosa  
## 38 4.9 3.6 1.4 0.1 setosa  
## 39 4.4 3.0 1.3 0.2 setosa  
## 40 5.1 3.4 1.5 0.2 setosa  
## 41 5.0 3.5 1.3 0.3 setosa  
## 42 4.5 2.3 1.3 0.3 setosa  
## 43 4.4 3.2 1.3 0.2 setosa  
## 44 5.0 3.5 1.6 0.6 setosa  
## 45 5.1 3.8 1.9 0.4 setosa  
## 46 4.8 3.0 1.4 0.3 setosa  
## 47 5.1 3.8 1.6 0.2 setosa  
## 48 4.6 3.2 1.4 0.2 setosa  
## 49 5.3 3.7 1.5 0.2 setosa  
## 50 5.0 3.3 1.4 0.2 setosa  
## 51 7.0 3.2 4.7 1.4 versicolor  
## 52 6.4 3.2 4.5 1.5 versicolor  
## 53 6.9 3.1 4.9 1.5 versicolor  
## 54 5.5 2.3 4.0 1.3 versicolor  
## 55 6.5 2.8 4.6 1.5 versicolor  
## 56 5.7 2.8 4.5 1.3 versicolor  
## 57 6.3 3.3 4.7 1.6 versicolor  
## 58 4.9 2.4 3.3 1.0 versicolor  
## 59 6.6 2.9 4.6 1.3 versicolor  
## 60 5.2 2.7 3.9 1.4 versicolor  
## 61 5.0 2.0 3.5 1.0 versicolor  
## 62 5.9 3.0 4.2 1.5 versicolor  
## 63 6.0 2.2 4.0 1.0 versicolor  
## 64 6.1 2.9 4.7 1.4 versicolor  
## 65 5.6 2.9 3.6 1.3 versicolor  
## 66 6.7 3.1 4.4 1.4 versicolor  
## 67 5.6 3.0 4.5 1.5 versicolor  
## 68 5.8 2.7 4.1 1.0 versicolor  
## 69 6.2 2.2 4.5 1.5 versicolor  
## 70 5.6 2.5 3.9 1.1 versicolor  
## 71 5.9 3.2 4.8 1.8 versicolor  
## 72 6.1 2.8 4.0 1.3 versicolor  
## 73 6.3 2.5 4.9 1.5 versicolor  
## 74 6.1 2.8 4.7 1.2 versicolor  
## 75 6.4 2.9 4.3 1.3 versicolor  
## 76 6.6 3.0 4.4 1.4 versicolor  
## 77 6.8 2.8 4.8 1.4 versicolor  
## 78 6.7 3.0 5.0 1.7 versicolor  
## 79 6.0 2.9 4.5 1.5 versicolor  
## 80 5.7 2.6 3.5 1.0 versicolor  
## 81 5.5 2.4 3.8 1.1 versicolor  
## 82 5.5 2.4 3.7 1.0 versicolor  
## 83 5.8 2.7 3.9 1.2 versicolor  
## 84 6.0 2.7 5.1 1.6 versicolor  
## 85 5.4 3.0 4.5 1.5 versicolor  
## 86 6.0 3.4 4.5 1.6 versicolor  
## 87 6.7 3.1 4.7 1.5 versicolor  
## 88 6.3 2.3 4.4 1.3 versicolor  
## 89 5.6 3.0 4.1 1.3 versicolor  
## 90 5.5 2.5 4.0 1.3 versicolor  
## 91 5.5 2.6 4.4 1.2 versicolor  
## 92 6.1 3.0 4.6 1.4 versicolor  
## 93 5.8 2.6 4.0 1.2 versicolor  
## 94 5.0 2.3 3.3 1.0 versicolor  
## 95 5.6 2.7 4.2 1.3 versicolor  
## 96 5.7 3.0 4.2 1.2 versicolor  
## 97 5.7 2.9 4.2 1.3 versicolor  
## 98 6.2 2.9 4.3 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 100 5.7 2.8 4.1 1.3 versicolor  
## 101 6.3 3.3 6.0 2.5 virginica  
## 102 5.8 2.7 5.1 1.9 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 104 6.3 2.9 5.6 1.8 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 107 4.9 2.5 4.5 1.7 virginica  
## 108 7.3 2.9 6.3 1.8 virginica  
## 109 6.7 2.5 5.8 1.8 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 112 6.4 2.7 5.3 1.9 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 117 6.5 3.0 5.5 1.8 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 120 6.0 2.2 5.0 1.5 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 124 6.3 2.7 4.9 1.8 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 126 7.2 3.2 6.0 1.8 virginica  
## 127 6.2 2.8 4.8 1.8 virginica  
## 128 6.1 3.0 4.9 1.8 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 130 7.2 3.0 5.8 1.6 virginica  
## 131 7.4 2.8 6.1 1.9 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 134 6.3 2.8 5.1 1.5 virginica  
## 135 6.1 2.6 5.6 1.4 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 138 6.4 3.1 5.5 1.8 virginica  
## 139 6.0 3.0 4.8 1.8 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 143 5.8 2.7 5.1 1.9 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

str(iris)

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

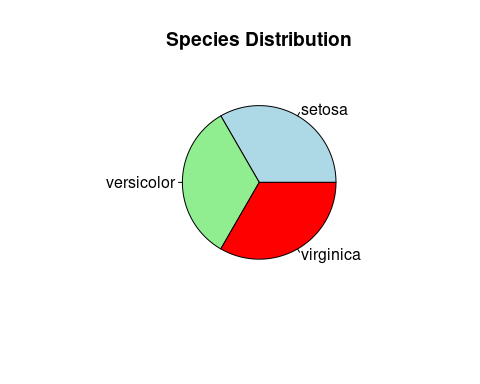
1. Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and petal.width. What is the R script and its result?

mean<- colMeans(iris[, c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")])  
mean

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## 5.843333 3.057333 3.758000 1.199333

1. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script and its result.

pie(table(iris$Species),  
 main = "Species Distribution",  
 labels = levels(iris$Species),  
 col = c("lightblue","lightgreen","red"))



#legend("topright", legend = levels(iris$Species), fill = c("lightblue", "lightgreen",  
# "red"),  
 # title = "Species Distribution")

1. Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last six (6) rows of each species.

setosaSub <- tail(subset(iris, Species == "setosa"), 6)  
versicolorSub <- tail(subset(iris, Species == "versicolor"), 6)  
virginicaSub <- tail(subset(iris, Species == "virginica"), 6)  
setosaSub

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 45 5.1 3.8 1.9 0.4 setosa  
## 46 4.8 3.0 1.4 0.3 setosa  
## 47 5.1 3.8 1.6 0.2 setosa  
## 48 4.6 3.2 1.4 0.2 setosa  
## 49 5.3 3.7 1.5 0.2 setosa  
## 50 5.0 3.3 1.4 0.2 setosa

versicolorSub

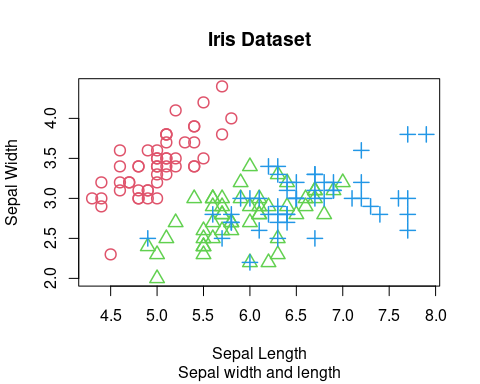
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 95 5.6 2.7 4.2 1.3 versicolor  
## 96 5.7 3.0 4.2 1.2 versicolor  
## 97 5.7 2.9 4.2 1.3 versicolor  
## 98 6.2 2.9 4.3 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 100 5.7 2.8 4.1 1.3 versicolor

virginicaSub

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

1. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versicolor,virginica). Add a title = “Iris Dataset”, subtitle = “Sepal width and length, labels for the x and y axis, the pch symbol and colors should be based on the species.

plot(iris$Sepal.Length, iris$Sepal.Width,  
pch = as.integer(iris$Species),  
col = as.integer(iris$Species) + 1,  
main = "Iris Dataset",  
sub = "Sepal width and length",  
xlab = "Sepal Length",  
ylab = "Sepal Width",  
cex = 1.5,  
lwd = 1.5)

 #the provided code successfully creates subsets for the Male and Female categories in the data set. Additionally, it demonstrates how to use R to analyze the data and interpret the results. The findings suggest that the data set may have more data on male individuals than female individuals and that the iris species can be distinguished by the length and width of their sepal and petal attributes.

1. Import the alexa-file.xlsx. Check on the variations. Notice that there are ex- tra whitespaces among black variants (Black Dot, Black Plus, Black Show, Black

Spot). Also on the white variants (White Dot, White Plus, White Show, White Spot). a. Rename the white and black variants by using gsub() function.

install.packages("readxl")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)

library(readxl)  
alexa <- read\_excel("alexa\_file.xlsx")  
print(alexa)

## # A tibble: 3,150 × 5  
## rating date variation verified\_reviews feedback  
## <dbl> <dttm> <chr> <chr> <dbl>  
## 1 5 2018-07-31 00:00:00 Charcoal Fabric Love my Echo! 1  
## 2 5 2018-07-31 00:00:00 Charcoal Fabric Loved it! 1  
## 3 4 2018-07-31 00:00:00 Walnut Finish Sometimes while play… 1  
## 4 5 2018-07-31 00:00:00 Charcoal Fabric I have had a lot of … 1  
## 5 5 2018-07-31 00:00:00 Charcoal Fabric Music 1  
## 6 5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo … 1  
## 7 3 2018-07-31 00:00:00 Sandstone Fabric Without having a cel… 1  
## 8 5 2018-07-31 00:00:00 Charcoal Fabric I think this is the … 1  
## 9 5 2018-07-30 00:00:00 Heather Gray Fabric looks great 1  
## 10 5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I’ve listen… 1  
## # ℹ 3,140 more rows

alexaVaration <- gsub("Black Plus", "Black Plus", alexa$variation)  
alexa$variation <- gsub("Black Show", "Black Show", alexa$variation)  
alexa$variation <- gsub("Black Spot", "Black Spot", alexa$variation)  
alexa$variation <- gsub("Black Dot", "Black Dot", alexa$variation)  
alexa$variation <- gsub("White Dot", "White Dot", alexa$variation)  
alexa$variation <- gsub("White Plus", "White Plus", alexa$variation)  
alexa$variation <- gsub("White Show", "White Show", alexa$variation)  
alexa$variation <- gsub("White Spot", "White Spot", alexa$variation)  
  
as.factor(iris$Species)

## [1] setosa setosa setosa setosa setosa setosa   
## [7] setosa setosa setosa setosa setosa setosa   
## [13] setosa setosa setosa setosa setosa setosa   
## [19] setosa setosa setosa setosa setosa setosa   
## [25] setosa setosa setosa setosa setosa setosa   
## [31] setosa setosa setosa setosa setosa setosa   
## [37] setosa setosa setosa setosa setosa setosa   
## [43] setosa setosa setosa setosa setosa setosa   
## [49] setosa setosa versicolor versicolor versicolor versicolor  
## [55] versicolor versicolor versicolor versicolor versicolor versicolor  
## [61] versicolor versicolor versicolor versicolor versicolor versicolor  
## [67] versicolor versicolor versicolor versicolor versicolor versicolor  
## [73] versicolor versicolor versicolor versicolor versicolor versicolor  
## [79] versicolor versicolor versicolor versicolor versicolor versicolor  
## [85] versicolor versicolor versicolor versicolor versicolor versicolor  
## [91] versicolor versicolor versicolor versicolor versicolor versicolor  
## [97] versicolor versicolor versicolor versicolor virginica virginica   
## [103] virginica virginica virginica virginica virginica virginica   
## [109] virginica virginica virginica virginica virginica virginica   
## [115] virginica virginica virginica virginica virginica virginica   
## [121] virginica virginica virginica virginica virginica virginica   
## [127] virginica virginica virginica virginica virginica virginica   
## [133] virginica virginica virginica virginica virginica virginica   
## [139] virginica virginica virginica virginica virginica virginica   
## [145] virginica virginica virginica virginica virginica virginica   
## Levels: setosa versicolor virginica

1. Get the total number of each variations and save it into another object. Save the object as variations.RData. Write the R scripts. What is its result? Hint: Use the dplyr package. Make sure to install it before loading the package.

library(readxl)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

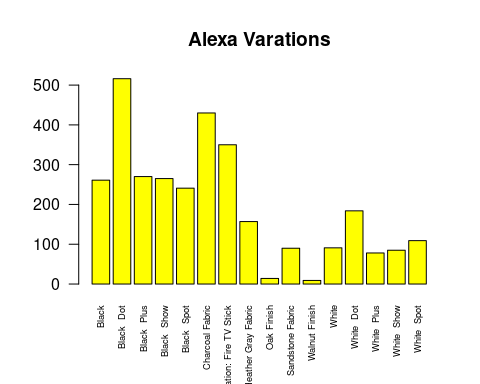
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

save(alexa, file = "variations.RData")  
load("variations.RData")  
  
alexaVar <- alexa%>%count(alexa$variation)  
alexaVar

## # A tibble: 16 × 2  
## `alexa$variation` n  
## <chr> <int>  
## 1 Black 261  
## 2 Black Dot 516  
## 3 Black Plus 270  
## 4 Black Show 265  
## 5 Black Spot 241  
## 6 Charcoal Fabric 430  
## 7 Configuration: Fire TV Stick 350  
## 8 Heather Gray Fabric 157  
## 9 Oak Finish 14  
## 10 Sandstone Fabric 90  
## 11 Walnut Finish 9  
## 12 White 91  
## 13 White Dot 184  
## 14 White Plus 78  
## 15 White Show 85  
## 16 White Spot 109

1. From the variations.RData, create a barplot(). Complete the details of the chart which include the title, color, labels of each bar.

barplot(  
height = alexaVar$n,  
names.arg = alexaVar$`alexa$variation`,  
col = "yellow",  
main = "Alexa Varations",  
las = 2,  
cex.names = 0.58  
)



1. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

par(mfrow = c(1, 2))  
  
black\_variants <- alexaVar[1:5,]  
white\_variants <- alexaVar[12:16,]  
  
barplot(  
 height = black\_variants$n,  
 names.arg = black\_variants$`alexa$variation`,  
 main = "Black Variants",  
 col = rainbow(5),  
 xlab = 'Total Numbers',  
 ylab = 'Frequency',  
 cex.names = 0.35,)  
  
barplot(  
 height = white\_variants$n,  
 names.arg = white\_variants$`alexa$variation`,  
 main = "White Variants",  
 col = rainbow(5),  
 xlab = 'Total Numbers',  
 ylab = 'Frequency',  
 cex.names = 0.35)

