RWorksheet_SOCO#4b

JanRay P. Soco

2025-11-24

```
#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.
vectorr <- c(1, 2, 3, 4, 5)
Matrix <- matrix(0, nrow = 5, ncol = 5)
for (i in 1:5) {
for (j in 1:5) {
Matrix[i, j] <- abs(i - j) + 1
}
}
Matrix
```

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

```
#2. Print the string "*" using for() function. The output should be the same as shown

for (i in 1:5) {
line <- ""
for (j in 1:i) {
line <- paste(line, "*")
}
print(line)
}
```

```
## [1] " *"
## [1] " * *"
## [1] " * * *"
## [1] " * * * *"
## [1] " * * * * *"
```

```
#3. 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.
num1 <- readline("Enter The Fibonachii number: ")
```

```
## Enter The Fibonachii number:
```

```
if (num1 == "") {
  start1 <- 10
} else {
  start1 <- as.numeric(num1)
}

if (is.na(start1)) {
  stop("Invalid input. Please enter a numeric value.")
}


a <- 0
b <- 1

repeat {
  fib <- a

  if (fib >= start1) {
    print(fib)
  }

  if (fib > 500) {
    break
  }

  temp <- a + b
  a <- b
  b <- temp
}
```

```
## [1] 13
## [1] 21
## [1] 34
## [1] 55
## [1] 89
## [1] 144
## [1] 233
## [1] 377
## [1] 610
```

```
#4. Import the dataset as shown in Figure 1 you have created previously.
#if csv
data_file <- read.csv("Householddata.csv")
head(data_file)
```

```
##   Respondents Sex Fathers_Occupation Persons_at_Home Siblings_at_School
## 1           1   2                  1               5                  6
## 2           2   2                  3               7                  4
## 3           3   1                  3               3                  4
## 4           4   2                  3               8                  1
## 5           5   2                  1               5                  2
## 6           6   2                  2               9                  1
##   Type_of_Houses
## 1              1
## 2              2
## 3              3
## 4              1
## 5              1
## 6              3
```

```
library(readxl)
household_data <- read_excel("Householddata.xlsx")
head(household_data)
```

```
## # A tibble: 6 x 6
##   Respondents   Sex Fathers_Occupation Persons_at_Home Siblings_at_School
##         <dbl> <dbl>              <dbl>           <dbl>              <dbl>
## 1           1     2                  1               5                  6
## 2           2     2                  3               7                  4
## 3           3     1                  3               3                  4
## 4           4     2                  3               8                  1
## 5           5     2                  1               5                  2
## 6           6     2                  2               9                  1
## # i 1 more variable: Type_of_Houses <dbl>
```

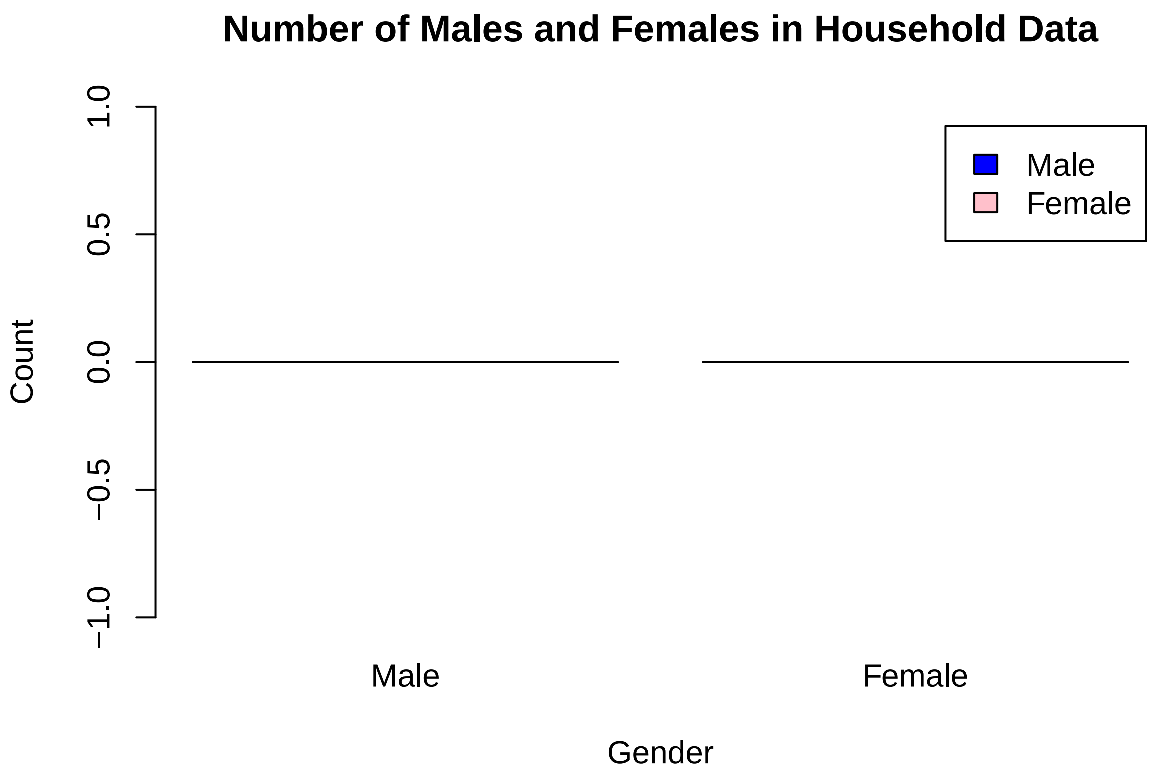
```
#b.

female_group <- subset(household_data, Sex == "F")
male_group <- subset(household_data, Sex == "M")
total_female <- nrow(female_group)
total_male <- nrow(male_group)
total_female
```

```
## [1] 0
```

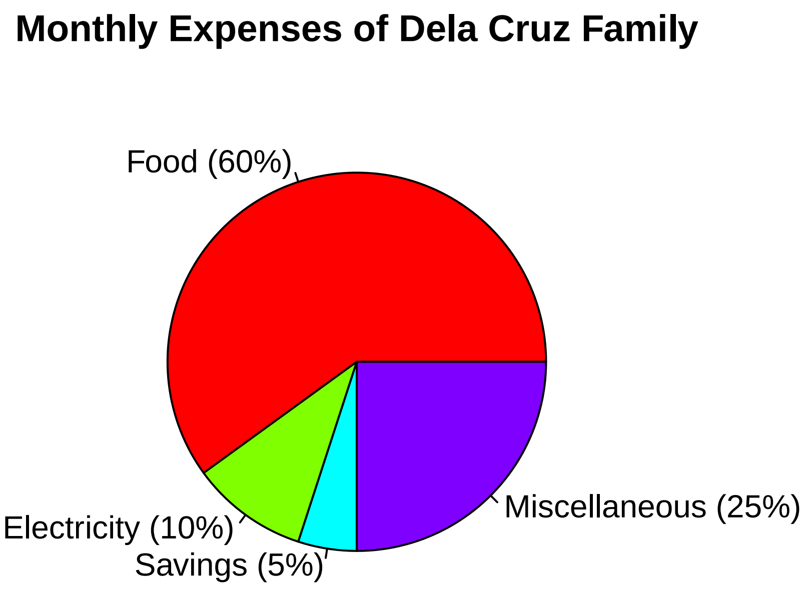
```
total_male
```

```
## [1] 0
```

```
#c.

gender_counts <- c(total_male, total_female)
barplot(
  gender_counts,
  names.arg = c("Male", "Female"),
  main = "Number of Males and Females in Household Data",
  xlab = "Gender",
  ylab = "Count",
  col = c("blue", "pink"),
  legend.text = c("Male", "Female")
)
```



```r
#5. The monthly income of Dela Cruz family was spent on the following:
#a.

expenses <- c(60, 10, 5, 25)
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")

percent_labels <- paste0(categories, " (", expenses, "%)")

pie(
  expenses,
  labels = percent_labels,
  main = "Monthly Expenses of Dela Cruz Family",
  col = rainbow(length(expenses))
)
```



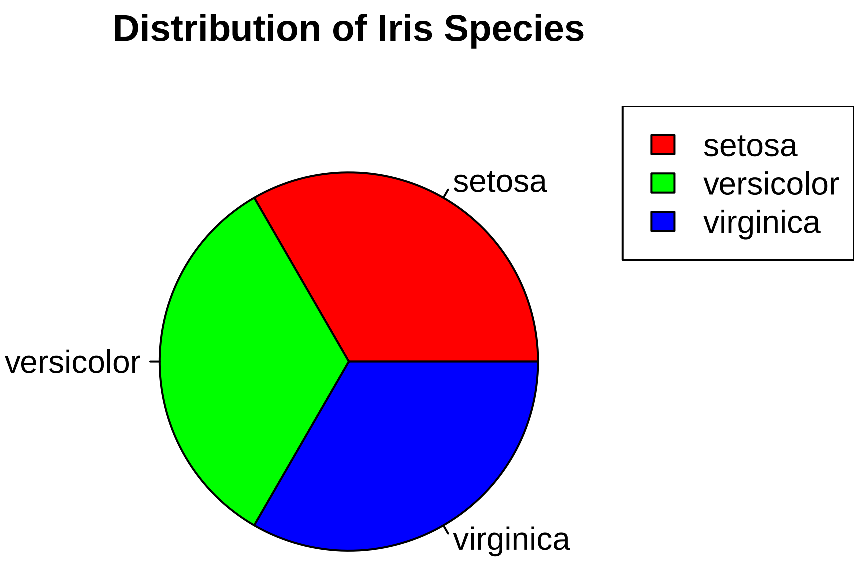
```r
#6. Use the iris dataset.
data(iris)
str(iris)
```

```
## '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 ...
```

```r
mean_values <- colMeans(iris[, 1:4])
mean_values
```

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

```r
species_counts <- table(iris$Species)
pie(
  species_counts,
  labels = names(species_counts),
  main = "Distribution of Iris Species",
  col = c("red", "green", "blue")
)
legend(
  "topright",
  legend = names(species_counts),
  fill = c("red", "green", "blue")
)
```



```r
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")
tail(setosa)
```

```
##    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
```

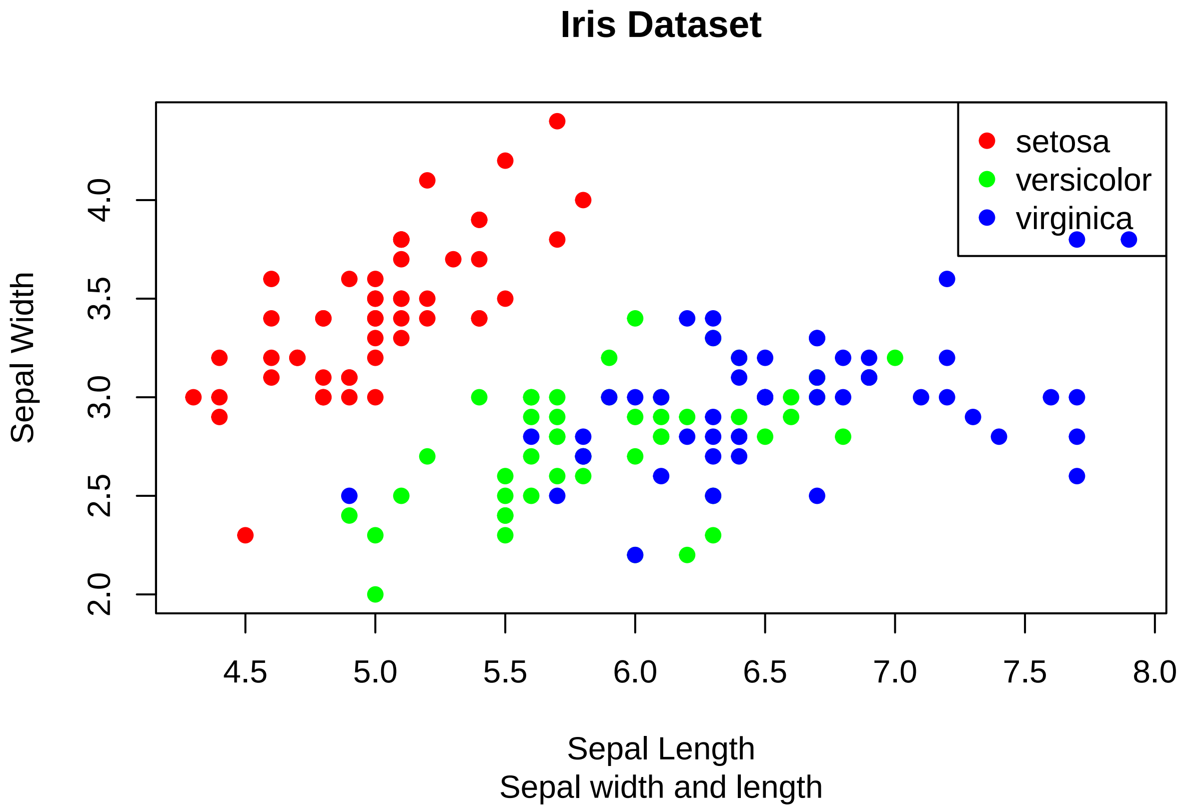
```r
tail(versicolor)
```

```
##     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
```

```r
tail(virginica)
```

```
##     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
```

```
#e. 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.

iris$Species <- as.factor(iris$Species)
colors <- c("red", "green", "blue")[iris$Species]
plot(
  iris$Sepal.Length,
  iris$Sepal.Width,
  col = colors,
  pch = 19,
  main = "Iris Dataset",
  sub = "Sepal width and length",
  xlab = "Sepal Length",
  ylab = "Sepal Width"
)

legend(
  "topright",
  legend = levels(iris$Species),
  col = c("red", "green", "blue"),
  pch = 19
)
```



```
#interpretation
```

```
#Setosa, shown in red, forms a clear and compact group with the widest sepal widths and the shortest se
```

```
#7. Import the alexa-file.xlsx. Check on the variations. Notice that there are extra whitespaces among
#a. Rename the white and black variants by using gsub() function.

library(readxl)
alexa <- read_excel("alexa_file.xlsx")

alexa$variation <- gsub("Black Dot",  "Black_Dot",  alexa$variation)
alexa$variation <- gsub("Black Plus", "Black_Plus", alexa$variation)
alexa$variation <- gsub("Black Show", "Black_Show", alexa$variation)
alexa$variation <- gsub("Black Spot", "Black_Spot", alexa$variation)

alexa$variation <- gsub("White Dot",  "White_Dot",  alexa$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)

head(alexa[, c("variation", "verified_reviews")])
```

```
## # A tibble: 6 x 2
##   variation           verified_reviews
##   <chr>               <chr>
## 1 Charcoal Fabric     Love my Echo!
## 2 Charcoal Fabric     Loved it!
## 3 Walnut Finish       Sometimes while playing a game, you can answer a question~
## 4 Charcoal Fabric     I have had a lot of fun with this thing. My 4 yr old lear~
## 5 Charcoal Fabric     Music
## 6 Heather Gray Fabric I received the echo as a gift. I needed another Bluetooth~
```

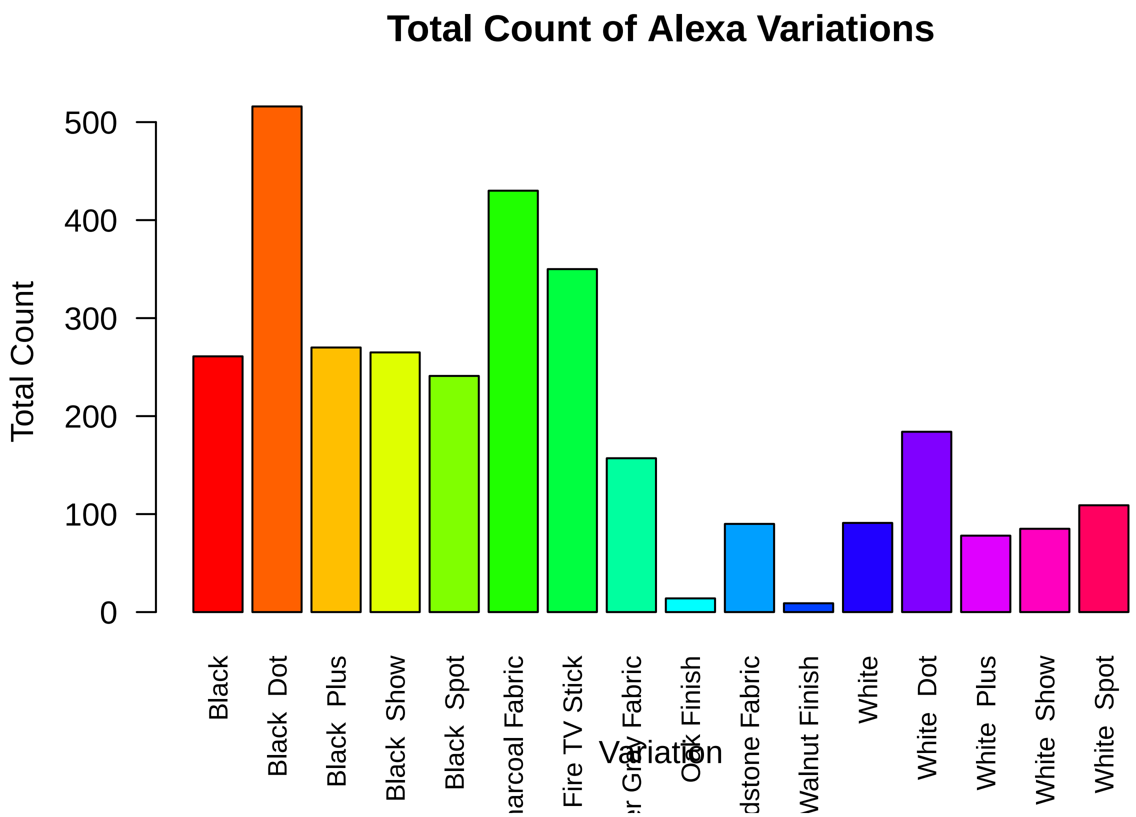
```
#b.
library(dplyr)
```

```
##
## 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
```

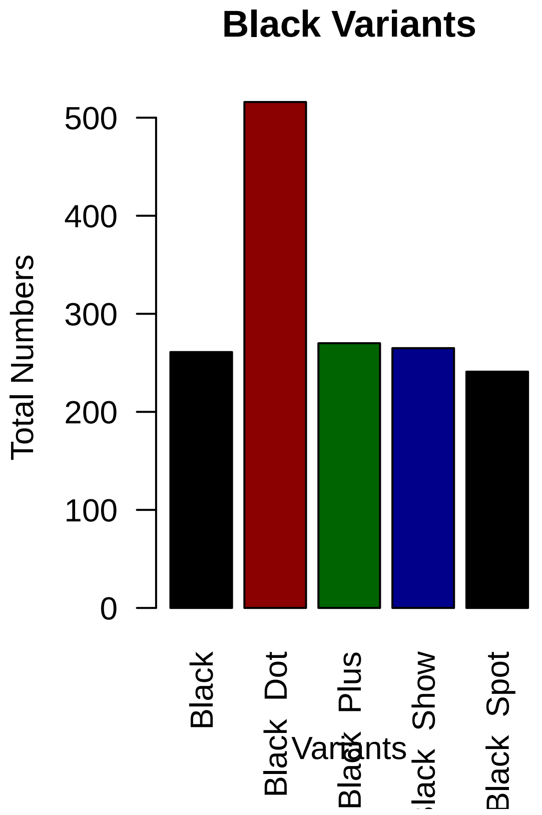
```
variation_counts <- alexa %>%
  count(variation)

variation_counts
```

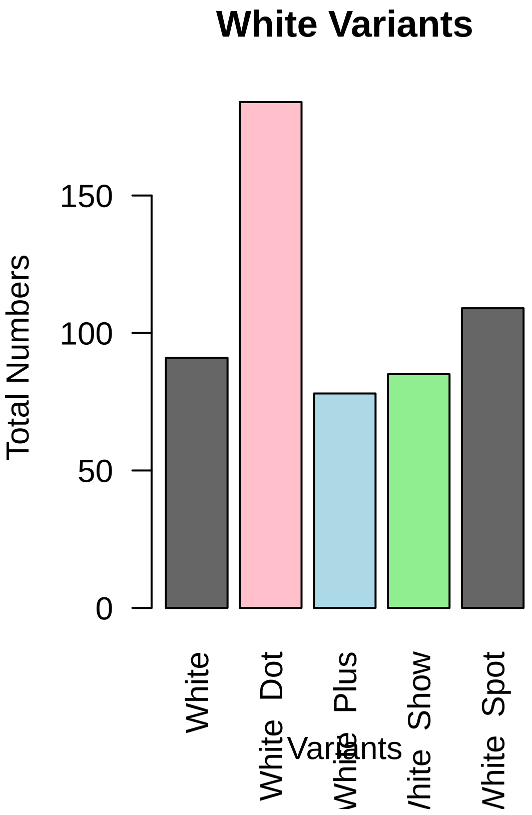
```
## # A tibble: 16 x 2
##    variation                        n
##    <chr>                          <int>
##  1 Black                            261
##  2 Black  Dot                       516
##  3 Black  Plus                      270
##  4 Black  Show                      265
##  5 Black  Spot                      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
```

```r
save(variation_counts, file = "variations.RData")


#C. Barplot
load("variations.RData")

barplot(
  variation_counts$n,
  names.arg = variation_counts$variation,
  col = rainbow(length(variation_counts$variation)),
  main = "Total Count of Alexa Variations",
  xlab = "Variation",
  ylab = "Total Count",
  las = 2,
  cex.names = 0.8
)
```



```r
#d.
black_variants <- subset(variation_counts, grepl("Black", variation))
white_variants <- subset(variation_counts, grepl("White", variation))

par(mfrow = c(1, 2))
barplot(
  black_variants$n,
  names.arg = black_variants$variation,
  col = c("black", "darkred", "darkgreen", "darkblue"),
  main = "Black Variants",
  xlab = "Variants",
  ylab = "Total Numbers",
  las = 2
)
barplot(
  white_variants$n,
  names.arg = white_variants$variation,
  col = c("gray40", "pink", "lightblue", "lightgreen"),
  main = "White Variants",
  xlab = "Variants",
  ylab = "Total Numbers",
  las = 2
)
```


Black Variants
Total Numbers
500
400
300
200
100
0
Black
Black Dot
Black Plus
Black Show
Black Spot
Variants


White Variants
Total Numbers
150
100
50
0
White
White Dot
White Plus
White Show
White Spot
Variants