Homework 1

Welcome to the first homework assignment!

The purpose of this homework is to gain experience using R and R Markdown, to review concepts from Introductory Statistics, to practice analyzing and plotting data in R. Please fill in the appropriate R code and write answers to all questions in the answer section. Once you have completed the assignment, submit a compiled pdf with your answers to Gradescope by 11pm on Sunday September 10th.

If you need help with the homework, please attend the TA office hours which are listed on Canvas and/or ask questions on Ed Discussions. Also, if you have completed the homework, please help others out by answering questions on Ed Discussions, which will count toward your class participation grade.

With this and all homework assignments, please be sure to knit your document often. This will help catch any mistakes right away so you will know where the mistake was made. If you don't knit often, it will be much harder to find any knitting errors and fix them!

Part 1: R Markdown practice

As we have discussed in class, R Markdown is a great way to create reproducible data analyses. To gain practice using R Markdown, all homework problem sets and your final project will be done in R Markdown.

R Markdown has a number of features that allow the text in your written reports to have better formatting. A cheatsheet for R Markdown formatting can be found here. When answering the questions be sure to knit your R Markdown document very often to catch errors as soon as they are made.

Part 1.1 (5 points): R Markdown fomatting

Please modify the lines of text below to change their formatting as described:

Make this line bold

Make this line italics

Make this line a second level header

- Make this line a bullet point

Make this text a hyperlink to yale.edu

Part 1.2 (5 points): LaTeX symbols

Please use LaTeX to write Plato's name in Greek below. An app to find LaTeX characters is here available at http://detexify.kirelabs.org/classify.html, or you can use Google.

Note: make sure the ending dollar sign touches the last letter otherwise you will get an error when knitting.

$\Pi \ldots$

$\Pi\lambda\alpha\tau\omega\nu$

Part 2: Using R within R Markdown documents

Let's now practice doing some basic computations in R. As described in class, all code in the R Markdown chunks is executed and the results are shown in the compiled output document (i.e., the code and output will be shown in the compiled pdf document).

Part 2.1 (5 points): Number journey

Please complete the following steps in the R Markdown chunk below:

1. Create an object called `a` and assign the value of 3 to it.
2. Create an object called `x` and assign the value 5 to it.
3. Create an object called `k` and assign the value 50 to it.
4. Create an object called `y` which has the value: $y = a \cdot 10^x + k$.
5. Print the value of y in the R chunk below.

In the answer section below, please write a couple of sentences on whether the object names `a`, `x`, `k` and `y` are good names to use and explain your reasoning.

```r
a <- 3
x <- 5
k <- 50
y <- a * 10 ^ x + k
y
```

```
## [1] 300050
```

Answer: [Describe whether the names `a`, `x`, `k` and `y` are good object names to use].

These are not good object names because they are vague and non-specific. The significance of the values assigned to `a`, `x`, `k`, and `y` is unclear, and if these variables were used regularly it would be easy to forget which variable was assigned to which value.

Part 2.2 (5 points): Summing elements in a vector

In the R chunk below, please create a vector called `my_vec` that has consecutive integers from 1 to 50; i.e., a vector of length 50 that has the numbers 1, 2, 3, . . . , 50 (hint: use a colon to create this vector rather than the `c()` function). Then add all the numbers in this vector together and print the result. Finally, check that you have the right answer using Gauss' formula for the sum of consecutive integers which is $S = \frac{n(n+1)}{2}$, where n = 50 here.

```
my_vec <- 1:50

# Sum integers from 1 to 50 directly

sum(my_vec)
```

```
## [1] 1275
```

```
# Use Gauss' formula
n <- 50
S <- (n * (n + 1)) / 2
S
```

```
## [1] 1275
```

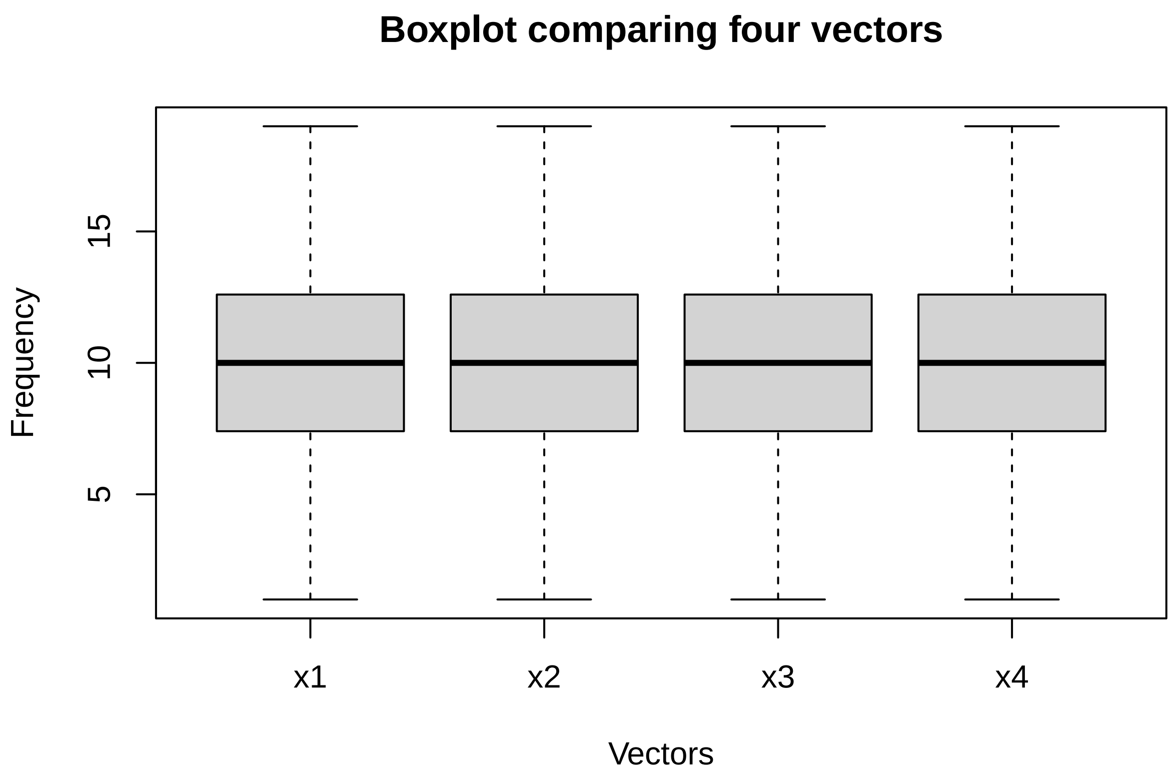
Part 3: Descriptive statistics and plots for quantitative data

In the following exercises, you will create and compare a few plots of quantitative data. Please answer each question, and if you notice any outliers in your data please address them appropriately. Also, be sure to put meaningful labels on your axes and add titles on all plots.

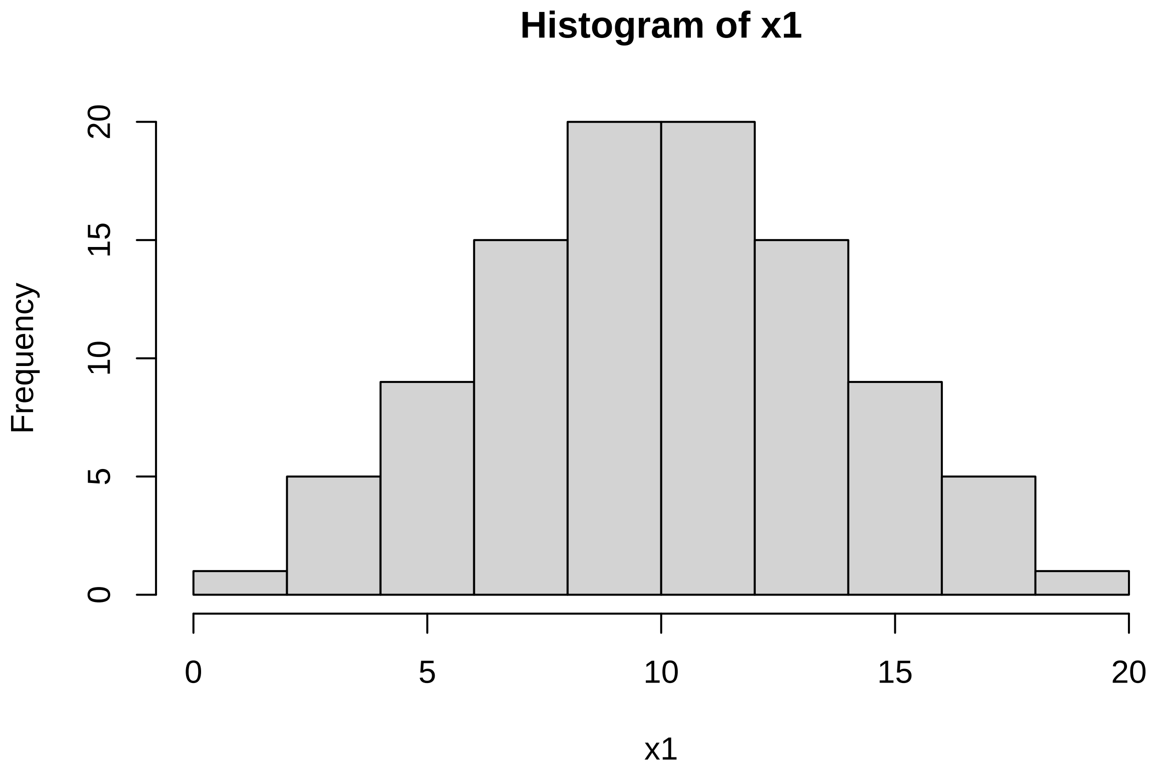
Part 3.1 (10 points)

The code chunk below loads four vector objects named `x1`, `x2`, `x3`, and `x4`. Create a side-by-side boxplot that compares these four vectors. Also create a histogram for each of these vectors (4 histograms total). Describe below whether the box plots or histograms are more informative for plotting this data and why.

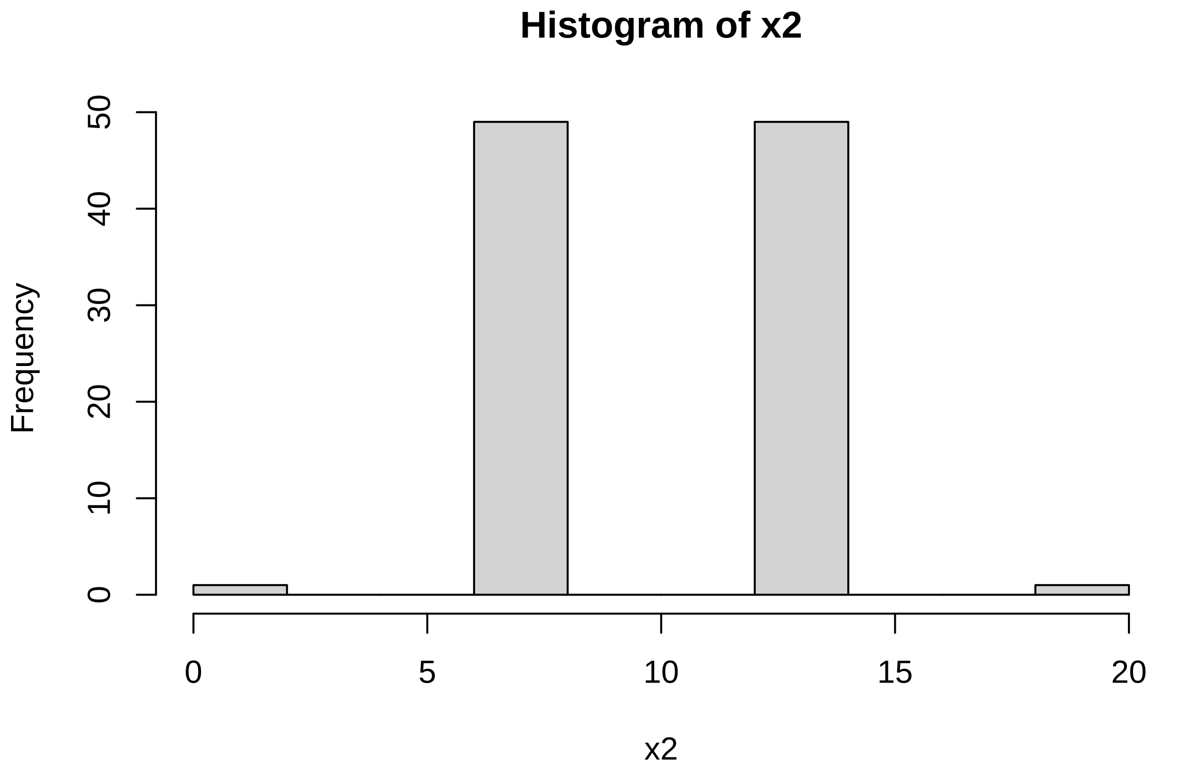
```r
load("misc_data.Rda")

boxplot(x1, x2, x3, x4,
        xlab = "Vectors",
        ylab = "Frequency",
        main = "Boxplot comparing four vectors",
        names = c("x1", "x2", "x3", "x4"))
```



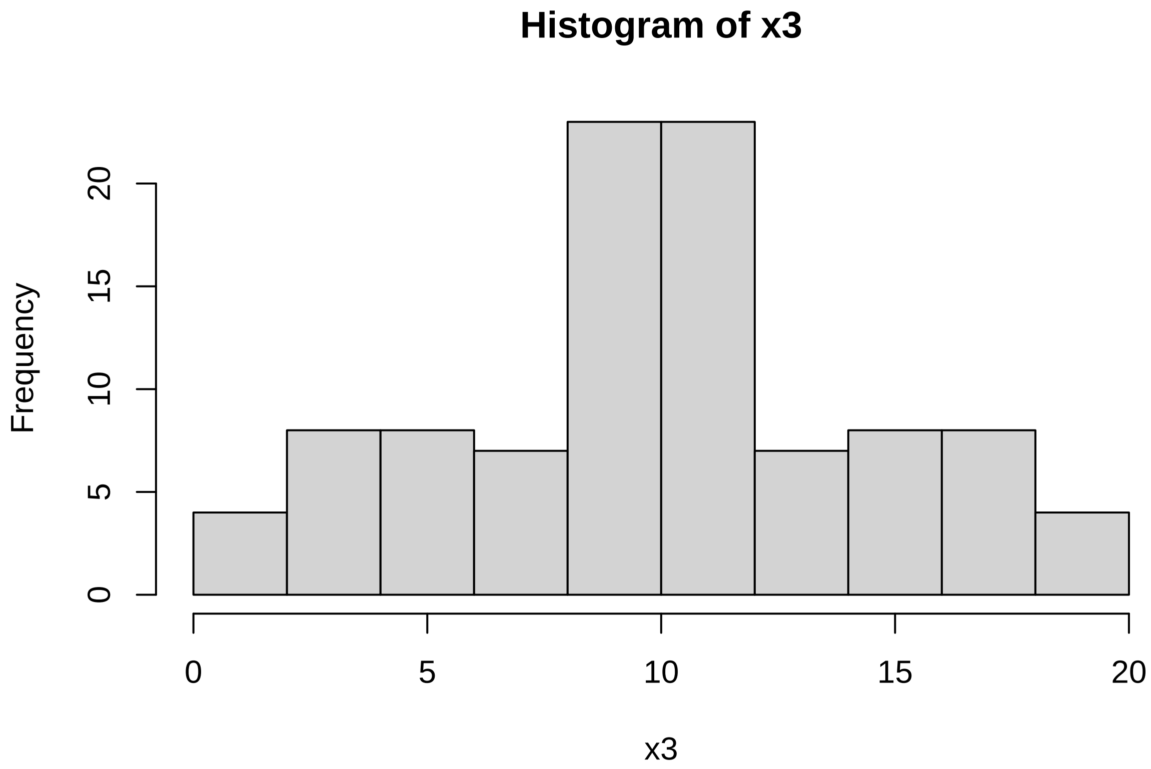
```r
hist(x1)
```



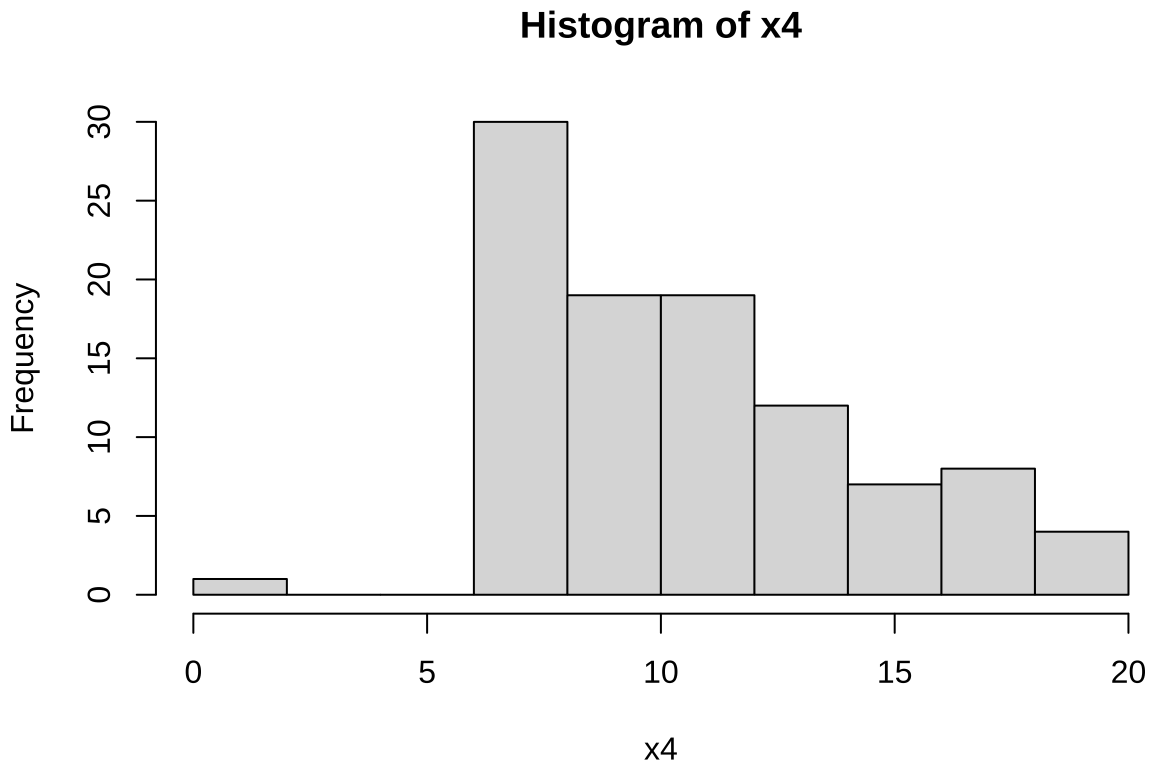
```
hist(x2)
```



```r
hist(x3)
```



```
hist(x4)
```



Answer: [Describe whether boxplots or histograms are more informative here]

Histograms are more effective for showing this data because they show more details about how the data is distributed for each vector. Boxplots show the first and third quartiles and the minimum, maximum, and median for a data set. It appears that in this case those values are equal for all four vectors, so their boxplots are the same. This is not as informative as the histograms, which demonstrate more clearly the differences between the vectors.

Part 3.2: (12 points)

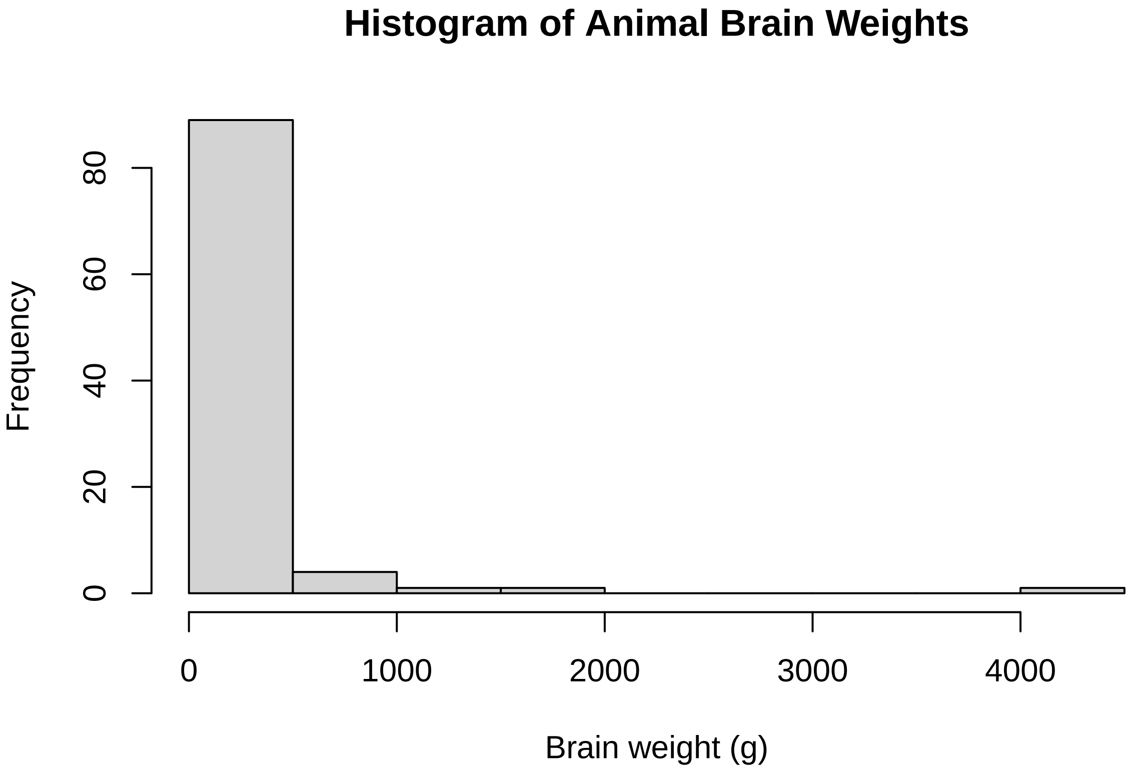
The R chunk below loads a data frame called `animals` which has information on 96 animals. The variables in this data frame are:

1. `Species`: name of the species
2. `Brain`: brain weight in grams
3. `Body`: body weight in kilograms
4. `Gestation`: gestation period in days
5. `Litter`: litter size

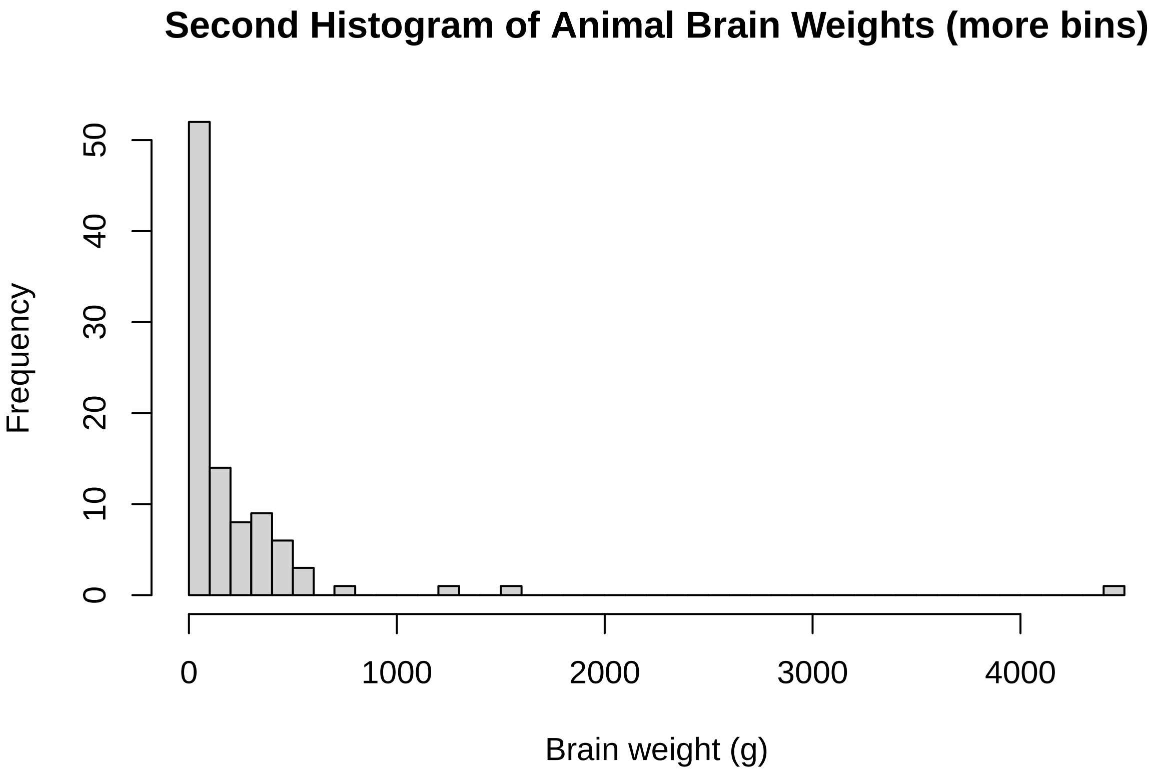
Please create a vector object that is called `brain_weight`, that has just the weights of the animals' brains. Then create a histogram and a boxplot of these brain weights using this vector object. In the answer section below, describe the shape of the distribution and investigate any outliers in the data; i.e., what is causing these outliers and are they surprising? Also describe the advantages that each type of plot has for showing features in the data.

Hint: for this problem you can inspect the data using RStudio's visual displays of data frames. In the future, it will be good to answer all questions by writing code that can show how you came to your conclusions.

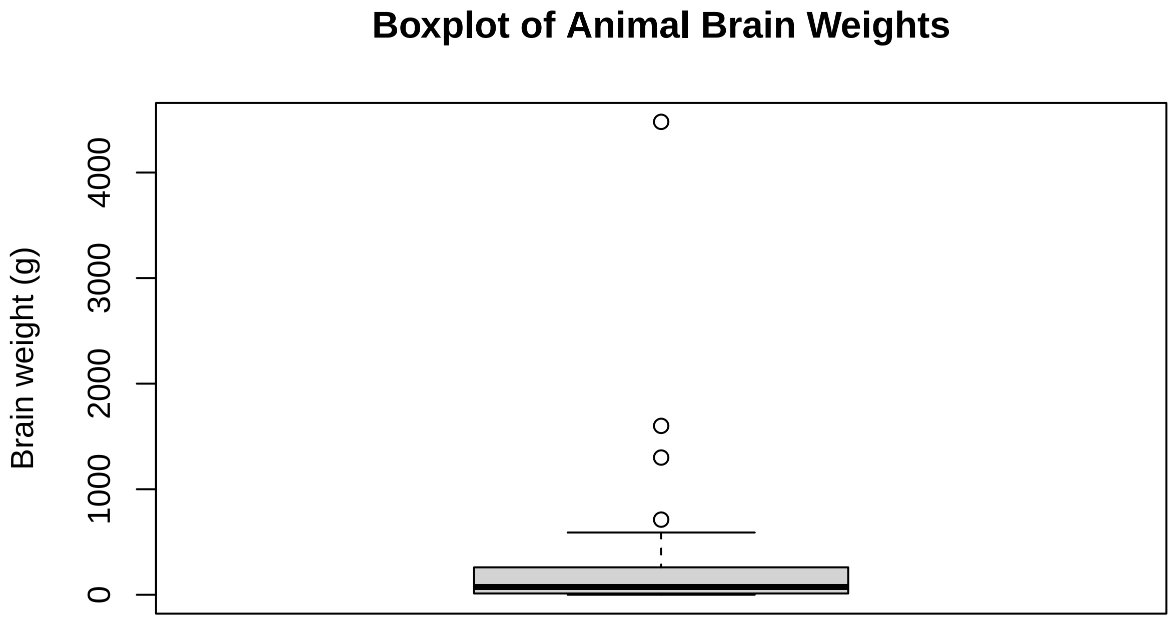
```
load("animals.rda")


# If you would like examine the data set you can type View(animals) on the
# console. However, do not include the View() function in your RMarkdown
# document since it will make the document fail to knit.


# Continue with your code below...

# Vector of animals' brain weights
brain_weight <- animals$Brain

# Histogram of animals' brain weights
hist(brain_weight, xlab = "Brain weight (g)", main = "Histogram of Animal Brain Weights")
```



```
hist(brain_weight, breaks = 50, xlab = "Brain weight (g)",
     main = "Second Histogram of Animal Brain Weights (more bins)")
```



```r
# Boxplot of animals' brain weights
boxplot(brain_weight, ylab = "Brain weight (g)", main = "Boxplot of Animal Brain Weights")
```



```
  # Boxplot shows outliers

# Identifying outliers
animals$Species[animals$Brain > 1000]
```

```
## [1] African elephant Dolphin          Human being
## 96 Levels: Aardvark Acouchis African elephant Agoutis Axis deer ... Yak
```

```
  # Outliers are animals which are known for their intelligence
```

Answer: [Describe advantages of boxplots and histograms for this data and investigate any unusual features of the data. 1-2 paragraphs with ~4-8 sentences total should suffice].

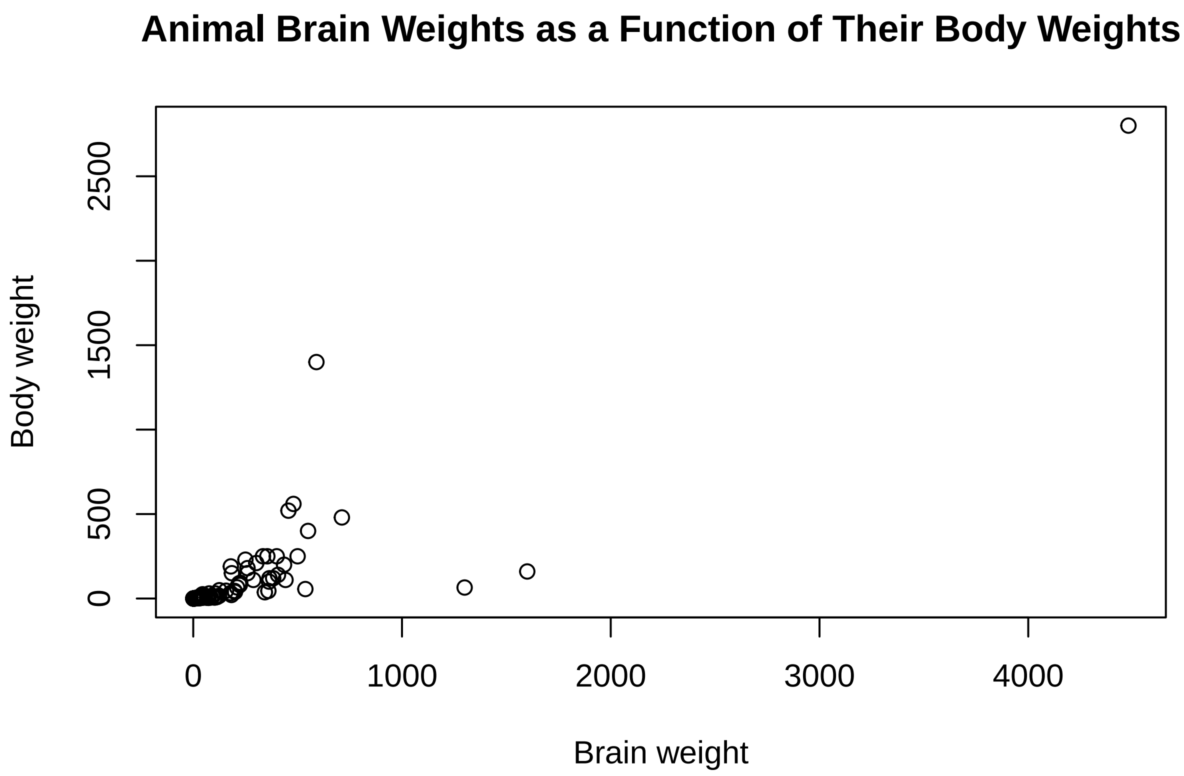
For this data, histograms are useful for showing the distribution of animals' brain weights. The first histogram shows that most animals have a brain weighing 500 grams or less, and very few have a brain weighing over 100 grams. Since '500 grams or less' is still a broad range, it may also be useful to add more bins to the graph. Doing so reveals that most animals have a brain weighing 100 grams or less, with some animals having brains weighing between 100 and 600 grams and very few having brains weighing brains weighing more than 600 grams.

Boxplots are useful for identifying outliers. Here the boxplot shows three major outliers. Filtering the data for brains weighing more than 1000 grams shows that these three animals are African elephants, dolphins, and human beings. Since elephants are known for their size and all three species are known for their intelligence, it is reasonable to say that these outliers do not represent an error in measurement and do not need to be removed from the data.

Part 3.3: (10 points)

Now create a scatter plot of the animals' brain weight as a function of their body weight. Describe what the results show, if the results are what you expected, and any limitations of the plot.

```
# Vector of animals' body weights
body_weight <- animals$Body

# Scatter plot of brain weight vs body weight
plot(brain_weight, body_weight, xlab = "Brain weight", ylab = "Body weight", main = "Animal Brain Weight
```



```
# Body weight outliers
animals$Species[animals$Body > 1000]
```

```
## [1] African elephant Hippopotamus
## 96 Levels: Aardvark Acouchis African elephant Agoutis Axis deer ... Yak
```

Answer:

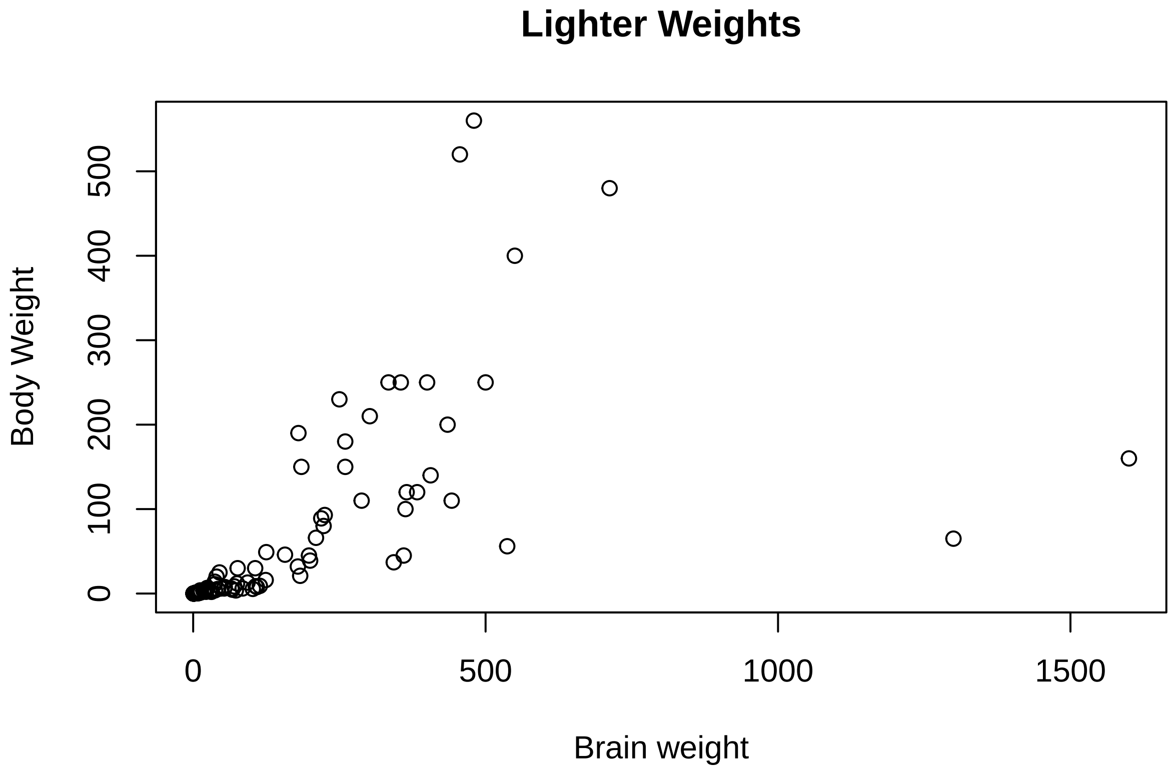
The scatter plot appears to show that there is at least a slight correlation between brain weight and body weight. Exceptions include humans and dolphins, which have a small body weight relative to their brain weight, and hippopotamuses, which have a large body weight relative to their brain weight. This is about what I expected, but since elephants represent an outlier for both brain and body weight, most of the points on the scatter plot are clustered closely together. This makes it is difficult to estimate the strength of the correlation between brain and body weight.

Part 3.4: (10 points)

Now let's create a data frame called `lighter_animals` that only has animals with body weights of less than 1,000 killograms. Hint: to do this first create a vector of Booleans called `lighter_inds` that has values of `TRUE` for animals that weigh less than 1,000 killograms and values of `FALSE` for animals that weigh 1,000 or more killograms. Then use the `lighter_inds` vector to create the `lighter_animals` data frame.

Once you have the `lighter_animals` data frame recreate the plot of brain weight as a function of body weight. Then describe whether this new plot more clearly shows the relationship between brain and body weight.

```r
lighter_inds <- body_weight < 1000

lighter_animals <- body_weight[lighter_inds]

lighter_brains <- brain_weight[lighter_inds]

plot(lighter_brains, lighter_animals, xlab = "Brain weight",
     ylab = "Body Weight", main = "Lighter Weights")
```



Answer:

In the new plot, African elephants and hippopotamuses are removed, so it more clearly shows a positive correlation between brain weight and body weight, with humans and dolphins remaining as outliers.

Part 4: Descriptive statistics and plots for categorical data

Heavy metal (or simply metal) is a genre of rock music that developed in the late 1960s and early 1970s that is characterized by distortion, extended guitar solos, emphatic beats, and loudness, with lyrics and performances that are sometimes associated with aggression and machismo (see Wikipedia). To practice exploring and visualizing categorical data, let's examine which countries most heavy metal bands come from.

A data file with a list of heavy metal bands was posted on Kaggle, and the code below loads a modified version of this data into an R data frame called `metal`. According to the kaggle website, the variables in this data frame are:

1. `band_name`: band name
2. `fans`: how many fans the band has on the website
3. `formed`: when the band formed
4. `origin`: the country of origin on the band
5. `split`: when the band split
6. `style`: the styles of the band

Please use the data loaded below for the following exercises.

```r
load("metal_bands.rda")

# If you would like examine the data set you can type View(metal) on the
# console. However, do not include the View() function in your RMarkdown
# document since it will make the document fail to knit.
```

Part 4.1: (10 points)

Use the `table()` function to create an object called `metal_counts` that has the counts of how many metal bands come from each country. What country did the most metal bands come from and what proportion of bands came from that country? Note: for the sake of simplicity, you can ignore bands that come from multiple countries for all the following exercises.

Also create a bar plot and pie chart showing the counts of how many bands come from each country. How do these plots look? How could you make them better?

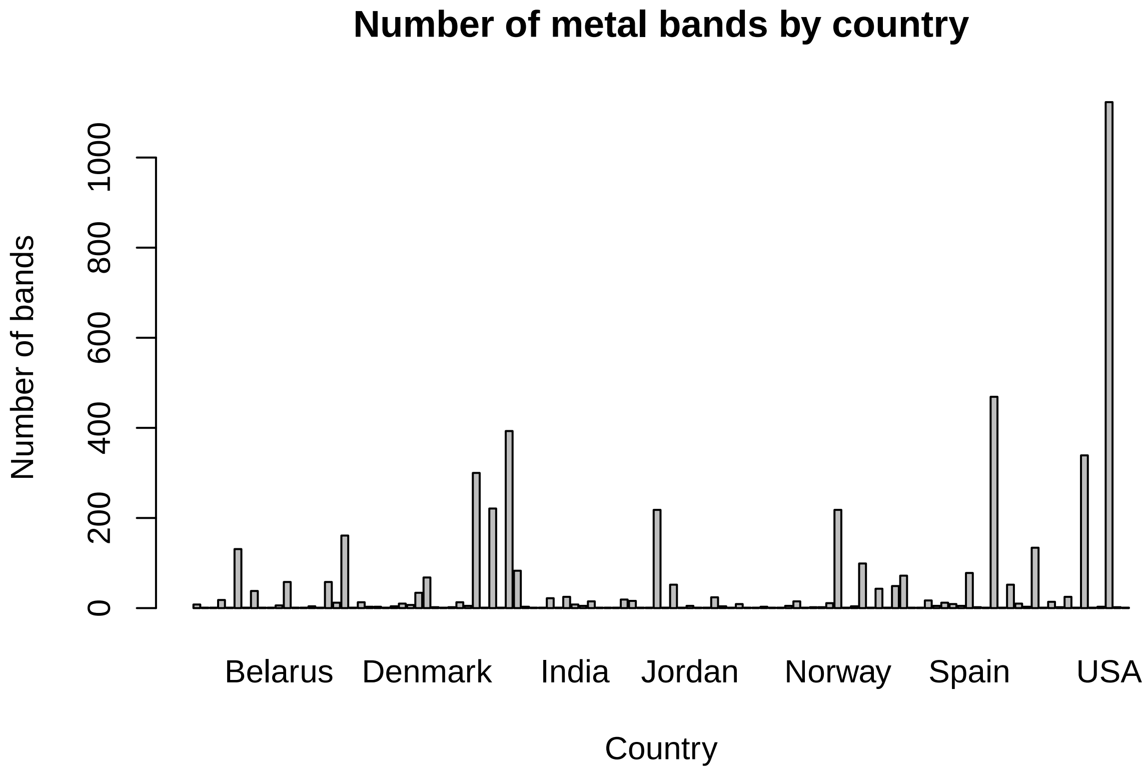
```r
# Table showing counts for how many metal bands come from each country
metal_counts <- table(metal$origin)

# Country with the most metal bands
most_counts <- metal_counts[metal_counts == max(metal_counts)]
most_counts
```

```
##  USA
## 1123
```

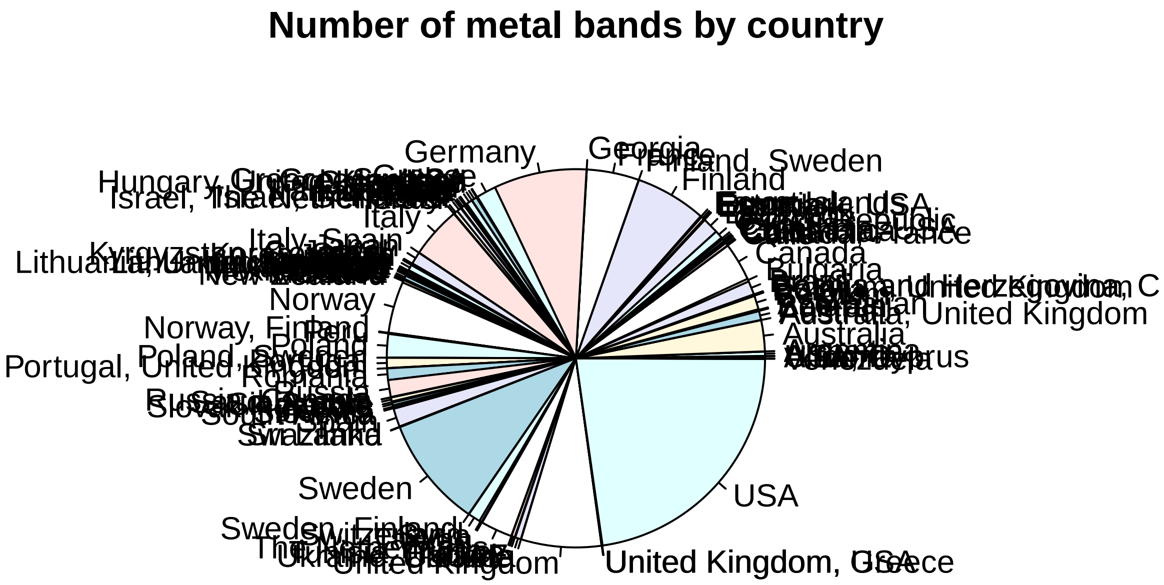
```
# Proportion of bands from country with the most metal bands
prop_counts <- prop.table(metal_counts) * 100
prop_most <- prop_counts[metal_counts == most_counts]
prop_most
```

```
##      USA
## 22.68687
```

```
# Bar plot showing counts of how many bands come from each country
barplot(metal_counts, xlab = "Country", ylab = "Number of bands", main = "Number of metal bands by coun
```



```
# Pie chart showing counts of how many bands come from each country
pie(metal_counts, main = "Number of metal bands by country")
```



Answers:

Both plots are difficult to interpret. The bar plot has only seven countries labeled, making it difficult to know which bar represents which country. The pie chart has every country labeled with most of the labels overlapping, making it difficult to tell which pie slice represents which country. In both cases the problem could be solved with a color-coded legend to replace the labels. Ideally this legend would be interactive so that users could click on a country name and see the corresponding bar or pie slice highlighted on the chart.

Alternatively, the charts could be simplified to show countries with few metal bands as a single bar or pie slice. This would make both charts easier to read.

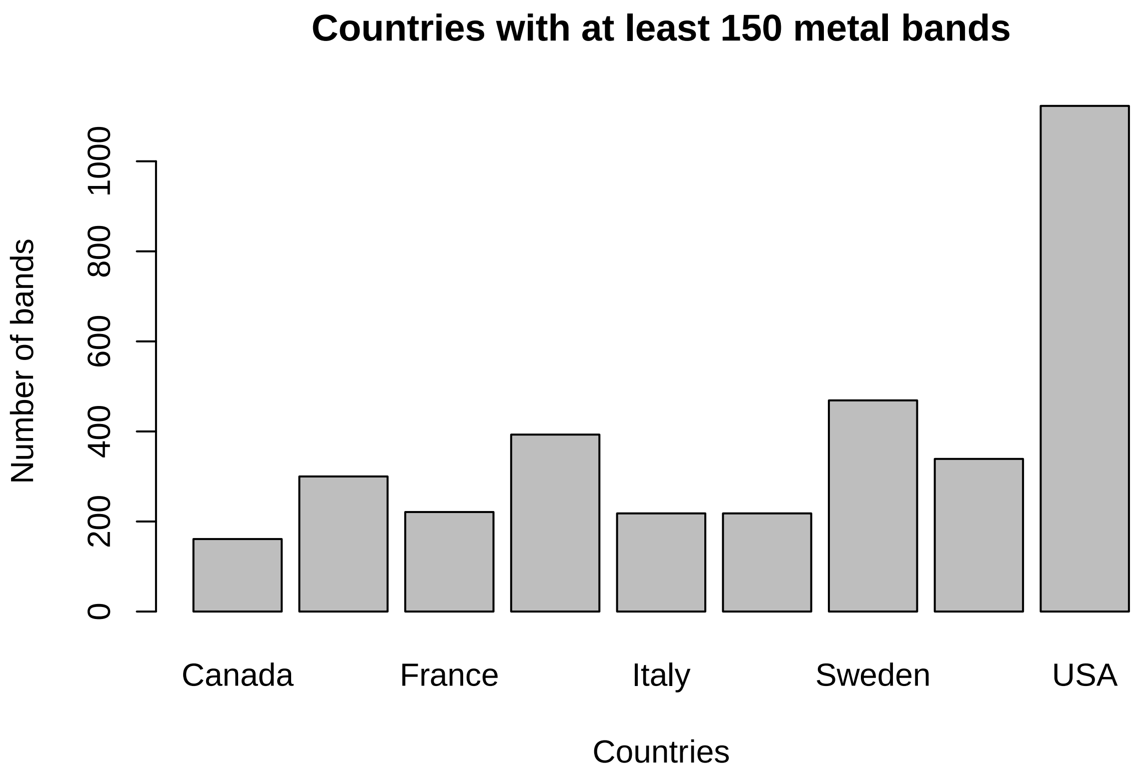
Part 4.2: (10 points)

Recreate your bar plot and pie chart to only show countries that have at least 150 bands originating from them (reviewing your answer to part 3.4 could be helpful). Also see if you can create a better color scheme for the pie chart. Hint: To figure out how to change the color of the pie chart, using `? pie` could be helpful. You might find it useful to look at this list of color names.

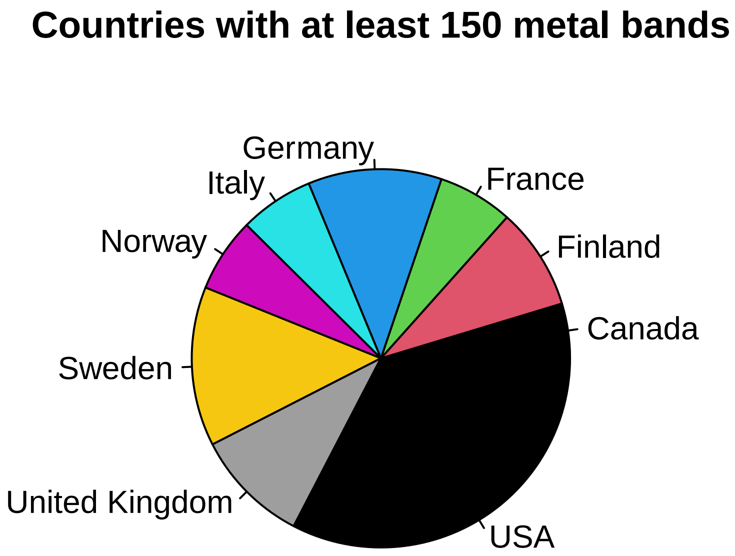
Bonus (0 points): see if you can create a version of the bar chart plotted horizontally where you can see all the country names.

```
# Countries that have at least 150 bands
more_bands <- metal_counts[metal_counts >=150]

# Bar plot with only countries that have at least 150 bands
```

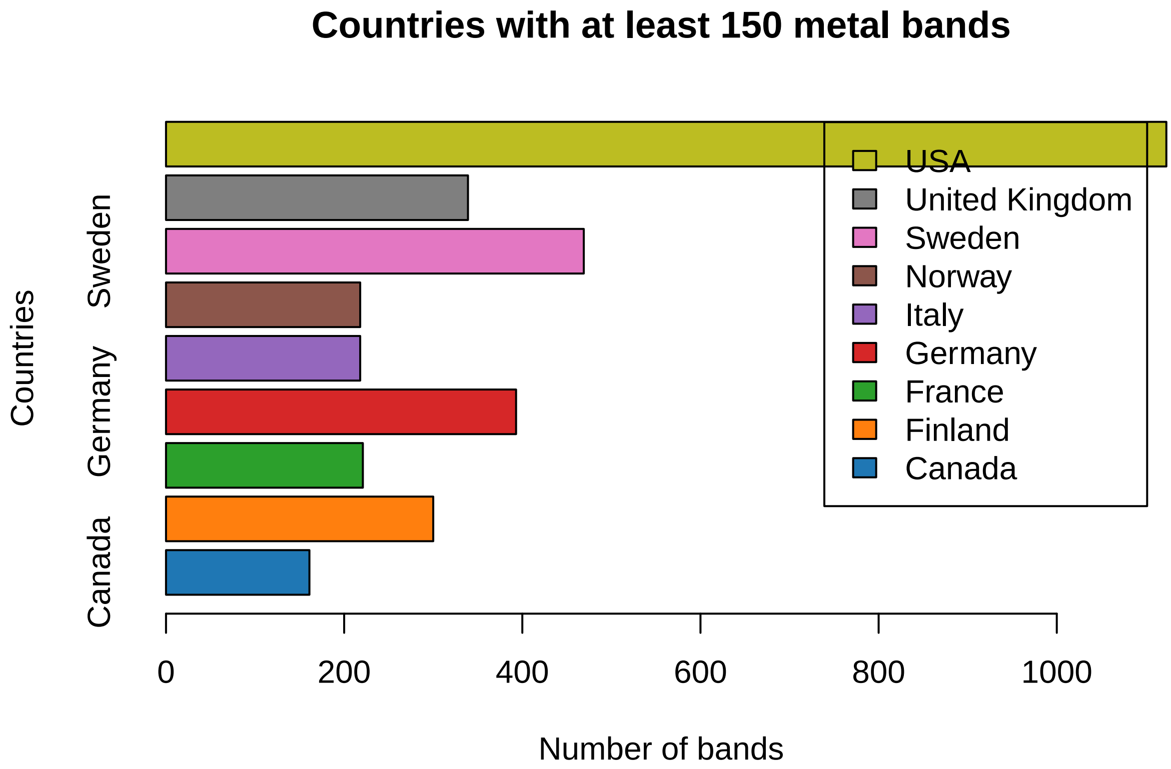
```r
barplot(more_bands, xlab = "Countries", ylab = "Number of bands",
        main = "Countries with at least 150 metal bands")
```



```r
# Pie chart with countries that have at least 150 bands
pie(more_bands, main = "Countries with at least 150 metal bands", col = palette("Classic Tableau"))
```



```r
# Bonus: Horizontal bar chart with all country names
barplot(more_bands, xlab = "Number of bands", ylab = "Countries", main = "Countries with at least 150 me
```



Part 4.3: (5 points) challenge problem

This is a "challenge problem" that you should try to figure out without getting help from the TAs. Challenges problems might be more difficult than other problems but they won't be worth too many points, so they will not have a large impact on your homework score.

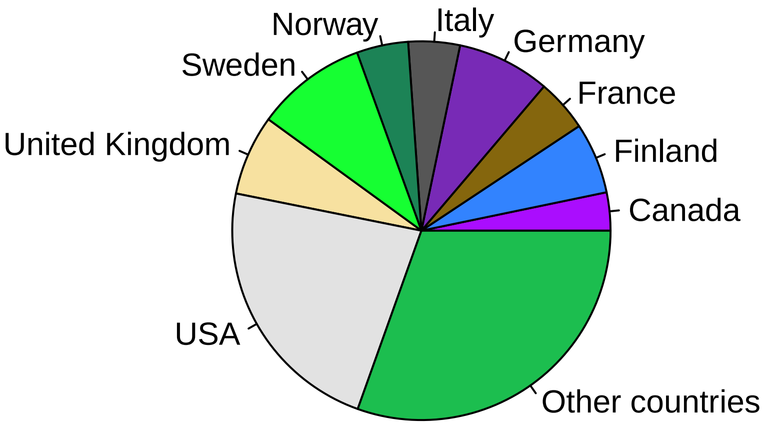
The plots in 4.2 are a bit misleading because they ignore all bands that come from many countries where only a few metal bands originated. The pie chart is particularly misleading because it gives the sense that particular countries have a high proportion of bands originating from them when this proportion is only out of the subset of all the countries in the original data set. A way to address this is to create a category called "other countries" that has the total number of bands from all countries that were left out in the original plot (i.e., the total number of bands from countries with less than 150 bands originating from them).

In the R chunk below, create a pie chart that has the category "other countries" which lists the total number of bands that originate from countries that were not included in figures you created in Part 4.2.

Hints:

1. If you have a vector (or table) called `my_vec`, you can concatenate the value 27 on to the vector `my_vec` using the syntax `my_vec <- c(my_vec, 27)`.
2. You can get the names of a named vector `my_vec` using the function `names(my_vec)` which returns a vector of strings. You can also assign names to a vector `my_vec` using the function `names(my_vec) <- vec_of_names`, where `vec_of_names` is a vector of character strings.

```r
# Pie chart with "other countries" category
other_countries <- sum(metal_counts[metal_counts < 150])

pie(c(more_bands, other_countries), labels = c(names(more_bands), "Other countries"),
    col = palette("Classic Tableau"))
```



Bonus problem (0 points)

If you prefer hip hop music over metal, another data set is loaded below that contains information on the number of unique words that different hip hop artists have used. The data comes from data world. The variables in this data frame are:

1. `words`: number of unique words in lyrical corpus
2. `era`: decade when first 35,000 lyrical words were released
3. `rapper`: rapper name
4. `id` dash separated name, unique for each artist

If you feel inspired, please play around with this data set and you can share any interesting findings or visualizations below. This question, however, is optional and worth no points toward your homework score.

For more interesting visualizations of this and related data see this pudding article

```
#load("hiphop_vocab.rda")
```

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 1

Figure 1: metal