Chapter 5: Logarithms, exponentials and data transformations

1. Consider the following dataset of positive values:

```r
x <- c(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024)
```

   These values increase exponentially – each one is double the previous.

   - Use the `log()` function in R to transform the data and store the result in a new variable.
   - Plot the original values (x) and the log-transformed values on a scatter plot (e.g. with `plot(x, log(x))`). What do you notice about the spacing between the points? How does the transformation affect the spread of the data?
   - Calculate the mean of the log-transformed values.
   - Then, calculate the mean of the original values, take the log of that mean, and compare it to the mean of the log-transformed data.
   - Which one is larger, 'the log of the average' or 'the average of the logs'? Why do you think that is?

   Hint: Log is a nonlinear transformation. Think about how taking the average before or after applying the transformation could lead to different results.

2. Use the built-in `cars` dataset in R, which contains the speed of cars and the distance required to stop:

   - Load the dataset with `data(cars)`.
   - Fit a linear model predicting stopping distance (`dist`) from speed.
   - Inspect the residuals using a histogram or a QQ plot. Do they look approximately normal?
   - Apply a log transformation to the stopping distance (`log(dist)`) and fit a new model.
   - Compare the fit and the residuals. How does the transformation affect the assumptions of the model? Why might a log transformation be useful here?

3. You are given a small dataset containing test scores for three students across three subjects: Math, English, and Science. Create the dataset using the code below:

```r
data_wide <- data.frame(Student = c("Alice", "Bob", "Charlie"),
                        Math = c(85, 90, 78),
                        English = c(88, 75, 89),
                        Science = c(92, 85, 84))
```

   - Use the `head()` function to inspect the dataset.
   - Reshape the data into *long format*, where each row corresponds to a single observation (i.e., one student's score in one subject).
   - The resulting dataset should have three columns: `Student`, `Subject`, and `Score`.

- Reflect on why long format is often preferred when analysing or visualizing data in R.

4. In this exercise, you will work with data from a classic experiment investigating the effects of caffeine on motor performance. Thirty male participants were randomly assigned to one of three groups and given either 0 mg, 100 mg or 200 mg of caffeine. Two hours later, their finger tapping speed was recorded as the number of taps per minute. Each group contains 10 participants.

    - Download the dataset `finger_tapping.csv` and load it into R using `read.csv("finger_tapping.csv")`.
    - The dataset is in *wide format*, with one column per caffeine dose. Convert it into *long format*, with two columns:
        - `dose`: indicating the caffeine dose group ('0', '100', or '200')
        - `taps`: the number of taps per minute.
    - Fit a linear model treating caffeine `dose` as a *numeric* predictor. What do the model coefficients suggest about the effect of caffeine on tapping speed?
    - Fit a second linear model treating `dose` as a *categorical* factor. Compare this model to the previous one. Which model provides a better fit?
    - Create a plot showing the mean tapping speed at each dose level. Add the predicted values from the model treating dose as continuous. How well does the linear model describe the pattern in the data?
    - Based on your analysis and the plot, does the effect of caffeine appear to be linear across doses?