Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 4**	
Title of Lab Assignment: Demonstrate data reduction and manipulation techniques.		
DOP: 13/10/2023	**DOS: 13/10/2023**	
CO Mapped: **CO3**	**PO Mapped:** **PO4, PO5, PO7, PO8, PO9,** **PSO1, PSO2**	**Signature:**

Practical No. 4

Aim: Implementation of Data reduction using subsetting, implementation and usage Dplyr & Tidyverse, select, transmute, arrange, filter, group-by on dataset.

Description:

Data reduction is a crucial step in data analysis, where you reduce the size or complexity of a dataset to focus on specific subsets of data, relevant variables, or to prepare data for further analysis. The `dplyr` and `tidyverse` packages in R offer powerful tools for data reduction, including functions like `select`, `transmute`, `arrange`, `filter`, and `group_by`.

Here's a breakdown of the mentioned functions in the context of data reduction:

1. **`select`:** This function allows you to choose specific columns from your dataset, thereby reducing the number of variables under consideration. By selecting only the columns of interest, you can make your dataset more manageable and relevant for your analysis.
2. **`transmute`:** While `select` allows you to choose columns, `transmute` is used to create new variables based on existing ones. You can compute new variables or transformations, which can be useful for summarization or further analysis. This can also help in reducing the dataset's dimensionality.
3. **`arrange`:** Sometimes it's necessary to reorder rows within your dataset, perhaps to examine data in a particular order. `arrange` is used to sort rows based on one or more variables, making it easier to visualize or analyze data that's in a specific order.
4. **`filter`:** Data reduction often involves filtering out rows that don't meet specific criteria. With `filter`, you can extract a subset of your data that fits a particular condition, reducing the dataset to only the relevant observations.
5. **`group_by`:** In some cases, data reduction is about aggregating data based on specific variables. `group_by` is used to create groups within your data based on a variable, allowing you to perform summary operations on these groups.

When performing data reduction, you're essentially focusing on specific aspects of your dataset that are relevant to your analysis objectives, thereby making your analysis more efficient and meaningful. Here's a summary of the steps involved in data reduction using these functions:

1. **Data Preparation:** Load your dataset and ensure it's in a suitable format for analysis.

2. **Select Relevant Columns:** Use `select` to choose the variables that are pertinent to your analysis. This reduces the dimensionality of your dataset.
3. **Create New Variables:** If necessary, use `transmute` to compute new variables or transformations that might be helpful for your analysis.
4. **Filter Data:** Use `filter` to subset your data based on specific conditions or criteria. This reduces the dataset to only the relevant observations.
5. **Arrange Data:** If the order of data is essential, use `arrange` to sort the data based on one or more variables.
6. **Group and Summarize Data:** If you need to aggregate your data, use `group_by` in combination with summarization functions to compute summary statistics for each group.

Code (Script):

```
# Load required libraries
library(dplyr)

# Load the mtcars dataset
data(mtcars)

# View the first few rows of the dataset
head(mtcars)

# Select specific columns
selected_columns <- mtcars %>%
  select(mpg, hp, wt)

# Filter rows based on conditions
filtered_data <- mtcars %>%
  filter(cyl == 6, gear == 4)

# Arrange rows based on a variable
arranged_data <- mtcars %>%
  arrange(desc(mpg))
```

```
# Group the data by a variable and summarize it
grouped_and_summarized <- mtcars %>%
  group_by(cyl) %>%
  summarize(mean_mpg = mean(mpg), mean_hp = mean(hp))

# Create new variables using transmute
transmuted_data <- mtcars %>%
  transmute(mpg_per_hp = mpg / hp, wt_miles_per_gallon = wt / mpg)

# View the resulting datasets
head(selected_columns)
head(filtered_data)
head(arranged_data)
grouped_and_summarized
head(transmuted_data)
```

Output:

```
> # Load the mtcars dataset
> data(mtcars)
>
> # View the first few rows of the dataset
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> # Select specific columns
> selected_columns <- mtcars %>%
+  select(mpg, hp, wt)
>
> # Filter rows based on conditions
> filtered_data <- mtcars %>%
+  filter(cyl == 6, gear == 4)
> # Arrange rows based on a variable
> arranged_data <- mtcars %>%
+  arrange(desc(mpg))
> # Group the data by a variable and summarize it
> grouped_and_summarized <- mtcars %>%
```

```
+  group_by(cyl) %>%
+  summarize(mean_mpg = mean(mpg), mean_hp = mean(hp))
> # Create new variables using transmute
> transmuted_data <- mtcars %>%
+  transmute(mpg_per_hp = mpg / hp, wt_miles_per_gallon = wt / mpg)
> # View the resulting datasets
> head(selected_columns)
                   mpg  hp    wt
Mazda RX4         21.0 110 2.620
Mazda RX4 Wag     21.0 110 2.875
Datsun 710        22.8  93 2.320
Hornet 4 Drive    21.4 110 3.215
Hornet Sportabout 18.7 175 3.440
Valiant           18.1 105 3.460
> head(filtered_data)
               mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Merc 280      19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
> head(arranged_data)
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
> grouped_and_summarized
# A tibble: 3 × 3
    cyl mean_mpg mean_hp
  <dbl>    <dbl>   <dbl>
1     4     26.7    82.6
2     6     19.7   122.
3     8     15.1   209.
> head(transmuted_data)
                  mpg_per_hp wt_miles_per_gallon
Mazda RX4          0.1909091           0.1247619
Mazda RX4 Wag      0.1909091           0.1369048
Datsun 710         0.2451613           0.1017544
Hornet 4 Drive     0.1945455           0.1502336
Hornet Sportabout  0.1068571           0.1839572
Valiant            0.1723810           0.1911602
```

Conclusion:

Here code demonstrates the effective use of dplyr and tidyverse functions for data reduction in R. By employing select, transmute, arrange, filter, and group_by, the code showcases how to reduce dataset size and complexity, focusing on specific data subsets and variables of interest. This data reduction process streamlines data analysis and improves the ability to extract valuable insights from the data.