

## PRACTICAL NO 5

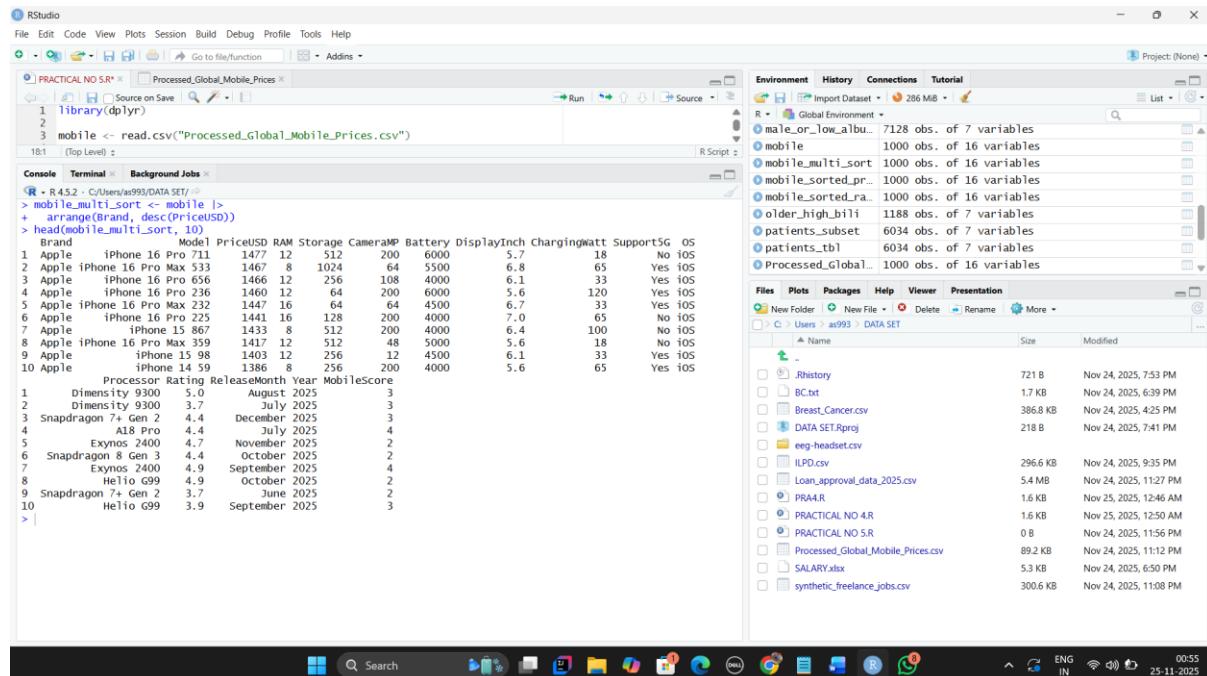
**AIM:-** Sorting data using `arrange()` in R.

## **OUTPUT:-**

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Source on Save, Run, Source.
- Project Bar:** Go to file/function, Addins, Project: (None).
- Code Editor:** PRACTICAL NO 5.R, showing R code for reading a CSV file and sorting mobile phones by price.
- Console:** Displays the output of the R code, including the resulting data frame.
- Data View:** Shows the 'mobile\_sorted\_price' data frame with columns: Brand, Model, PriceUSD, RAM, Storage, CameraMP, Battery, DisplayInch, ChargingWatt, Support5G, Rating, ReleaseMonth, Year, and MobileScore.
- Environment View:** Lists objects in the environment, including 'low\_interest\_fil...', 'male\_or\_low\_albu...', 'mobile', 'mobile\_sorted\_pr...', 'older\_high\_bili...', 'patients\_subset', 'patients\_tbl', and 'Processed\_Global...'. The 'mobile' object is described as having 1000 obs. of 16 variables.
- File Browser:** Lists files in the current directory, including 'Rhistory', 'BC.txt', 'Breast\_Cancer.csv', 'DATA\_SET.prj', 'eeg-headset.csv', 'ILPD.csv', 'Loan\_approval\_data\_2025.csv', 'PRAIR.R', 'PRACTICAL NO 4.R', 'PRACTICAL NO 5.R', 'Processed\_Global\_Mobile\_Prices.csv', 'SALARY.xlsx', and 'synthetic\_freelance\_jobs.csv'.

**Sheth L.U.J. College of Arts And Sir M.V. College of Science and Commerce**  
**Data Analysis with SAS / SPSS /R**



RStudio interface showing the initial steps of data analysis:

```

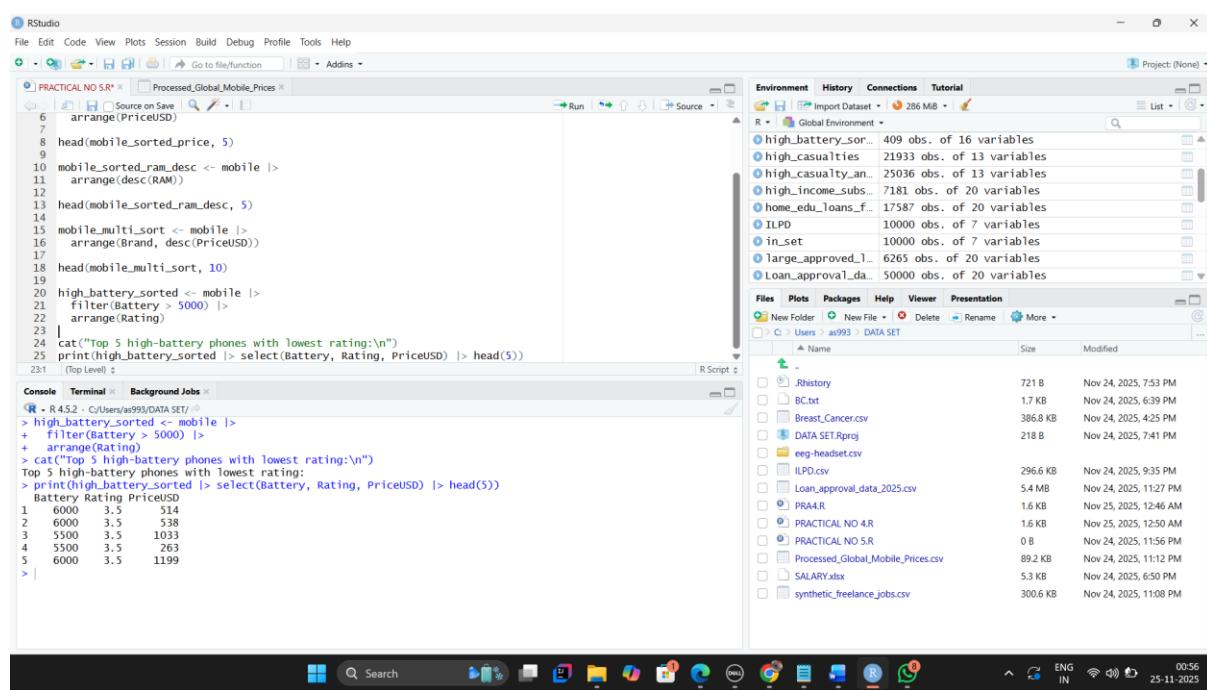
library(dplyr)
mobile <- read.csv("Processed_Global_Mobile_Prices.csv")
head(mobile_multi_sort, 10)
  
```

The console output shows the first 10 rows of the sorted mobile dataset:

Brand	Model	Processor	RAM	Storage	Camera	MP	Battery	Display	Inch	Charging	watt	Support	5G	OS
Apple	iPhone 16 Pro	711	1477	12	512	200	6000	5.7	18	No	iOS			
Apple	iPhone 16 Pro Max	713	1479	8	1024	200	5120	6.8	65	Yes	iOS			
Apple	iPhone 16 Pro	656	1466	12	256	108	4000	6.1	33	Yes	iOS			
Apple	iPhone 16 Pro	236	1460	12	64	200	6000	5.6	120	Yes	iOS			
Apple	iPhone 16 Pro Max	232	1447	16	64	64	4500	6.7	33	Yes	iOS			
Apple	iPhone 16 Pro	225	1441	16	128	200	4000	7.0	65	No	iOS			
Apple	iPhone 15	867	1433	8	512	200	4000	6.4	100	No	iOS			
Apple	iPhone 16 Pro Max	359	1417	12	512	48	5000	5.6	18	No	iOS			
Apple	iPhone 15	98	1403	12	256	12	4500	6.1	33	Yes	iOS			
Apple	iPhone 14	59	1386	8	256	200	4000	5.6	65	Yes	iOS			

Processor Rating ReleaseMo Year MobileScore

Processor	Rating	ReleaseMo	Year	MobileScore
Dimensys 9300	5.0	August 2025	3	
Dimensity 9300	3.7	July 2025	3	
Snapdragon 7+ Gen 2	4.4	December 2025	3	
A18 Pro	4.4	July 2025	4	
Exynos 2400	4.7	November 2025	2	
Snapdragon 8 Gen 3	4.4	October 2025	2	
Exynos 2400	4.9	September 2025	4	
Hello G99	4.9	October 2025	2	
Snapdragon 7+ Gen 2	3.7	June 2025	2	
Hello G99	3.9	September 2025	3	



RStudio interface showing the filtering and printing of high-battery phones:

```

arrange(PriceUSD)
head(mobile_sorted_price, 5)
mobile_sorted_ram_desc <- mobile |>
  arrange(desc(RAM))
head(mobile_sorted_ram_desc, 5)
mobile_multi_sort <- mobile |>
  arrange(Brand, desc(PriceUSD))
head(mobile_multi_sort, 10)
high_battery_sorted <- mobile |>
  filter(Battery > 5000) |>
  arrange(Rating)
cat("Top 5 high-battery phones with lowest rating:\n")
print(high_battery_sorted |> select(Battery, Rating, PriceUSD) |> head(5))
  
```

The console output shows the top 5 high-battery phones with the lowest rating:

Battery	Rating	PriceUSD
6000	3.5	514
6000	3.5	538
5500	3.5	1033
5500	3.5	263
6000	3.5	1199

