

## PRACTICAL NO 2

AIM: Generating frequency tables using table() or count() ®

The screenshot shows the RStudio interface with the following content:

```
R - R452 -  
> library(dplyr)  
> df <- read_csv("random_stock_market_dataset.csv")  
Rows: 60 Columns: 6  
Column specification:  
dbl (5): Open, High, Low, Close, Volume  
date (1): Date  
  
I use 'spec()' to retrieve the full column specification for this data.  
I specify the column types or set 'show_col_types = FALSE' to quiet this message.  
> # Frequency tables  
> open_counts <- table(df$open)  
> print("Frequency table: open Prices")  
[1] "Frequency table: Open Prices"  
> print(open_counts)  
112.68 114.16 120.34 147.85 155.92 167.91 170.55 186.19 187.88 190.11 195.75 197.41 210.9 216.9 221.48 223.45 231.07  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
231.79 241.35 245.47 253.13 256.08 257.18 258.01 267.93 278.68 283.33 289.9 296.45 302.73 303.75 314.61 327.4 330.34  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
340.76 354.97 355.67 358.7 368.41 369.78 372.44 379.51 380.43 390.1 395.02 399.54 406.91 414.61 430.58 446.59 451.06  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
455.69 462.88 464.4 466.44 467.37 468.27 469.61 481.91 492.79  
1 1 1 1 1 1 1 1  
  
> volume_counts <- table(df$volume)  
> print("Frequency table: volume")  
[1] "Frequency table: volume"  
> print(volume_counts)  
13193 13710 17174 17358 19354 20847 27539 28315 28704 28766 29000 36696 45928 47071 47481 48195 49334  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
55243 58091 62294 62443 64201 64993 67480 70326 79832 84859 89556 91831 93133 94568 95424 99680 106396  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
108022 112168 112415 115659 117694 122172 125917 126705 127026 129565 129955 133804 142032 145680 154376 156435 163847  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
167233 171229 179150 181314 183280 185680 188553 190216 195189  
1 1 1 1 1 1 1 1 1 1  
  
> close_counts <- table(df$close)  
> print("Frequency Table: Close Prices")  
[1] "Frequency table: Close Prices"  
> print(close_counts)  
121.27 129.95 130.29 151.34 166.49 174.37 178.58 191.4 193.36 198.37 205.89 210.2 220.31 226.35 234.46 238.17 242.69  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
245.42 252.2 258.95 260.41 260.49 264.94 272.14 275.57 285.22 291.79 298.01 303.72 305.95 312.82 328.56 334.93 346.79  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
351.46 369.1 370.56 373.51 373.96 375.29 375.67 384.57 395.51 397.68 413.27 414.04 417.6 421.92 435.79 452.93 461.23  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
465.41 472.06 472.23 475.35 479.63 480.22 481.98 491.99 500.42  
1 1 1 1 1 1 1 1  
  
> # Data Frame Format counts  
> open_df <- df %>% count(open)  
> print("Open Price Frequency (Data Frame Format)")  
[1] "Open Price Frequency (Data Frame Format)"  
> print(open_df)  
# A tibble: 60 x 2  
  open     n  
  <dbl> <int>  
1 113. 1  
2 114. 1  
3 120. 1  
4 148. 1  
5 156. 1  
6 168. 1  
7 171. 1  
8 186. 1  
9 188. 1  
10 190. 1  
# I 50 more rows  
# I use 'print(n = ...)' to see more rows  
> volume_df <- df %>% count(volume)  
> print("Volume Frequency (Data Frame Format)")  
[1] "Volume Frequency (Data Frame Format)"  
> print(volume_df)  
# A tibble: 60 x 2  
  volume     n  
  <dbl> <int>  
1 13193 1  
2 13710 1  
3 17174 1  
4 17358 1  
5 19354 1  
6 20847 1  
7 27539 1  
8 28315 1  
9 28704 1  
10 28766 1  
# I 50 more rows  
# I use 'print(n = ...)' to see more rows  
> close_df <- df %>% count(close)  
> print("Close Price Frequency (Data Frame Format)")  
[1] "Close Price Frequency (Data Frame Format)"  
> print(close_df)  
# A tibble: 60 x 2  
  close     n  
  <dbl> <int>  
1 121. 1  
2 130. 1  
3 130. 1  
4 151. 1  
5 166. 1  
6 174. 1  
7 179. 1  
8 191. 1
```

The screenshot shows the RStudio interface with the following content:

```
R - R452 -  
> # Data Frame Format counts  
> open_df <- df %>% count(open)  
> print("Open Price Frequency (Data Frame Format)")  
[1] "Open Price Frequency (Data Frame Format)"  
> print(open_df)  
# A tibble: 60 x 2  
  open     n  
  <dbl> <int>  
1 113. 1  
2 114. 1  
3 120. 1  
4 148. 1  
5 156. 1  
6 168. 1  
7 171. 1  
8 186. 1  
9 188. 1  
10 190. 1  
# I 50 more rows  
# I use 'print(n = ...)' to see more rows  
> volume_df <- df %>% count(volume)  
> print("Volume Frequency (Data Frame Format)")  
[1] "Volume Frequency (Data Frame Format)"  
> print(volume_df)  
# A tibble: 60 x 2  
  volume     n  
  <dbl> <int>  
1 13193 1  
2 13710 1  
3 17174 1  
4 17358 1  
5 19354 1  
6 20847 1  
7 27539 1  
8 28315 1  
9 28704 1  
10 28766 1  
# I 50 more rows  
# I use 'print(n = ...)' to see more rows  
> close_df <- df %>% count(close)  
> print("Close Price Frequency (Data Frame Format)")  
[1] "Close Price Frequency (Data Frame Format)"  
> print(close_df)  
# A tibble: 60 x 2  
  close     n  
  <dbl> <int>  
1 121. 1  
2 130. 1  
3 130. 1  
4 151. 1  
5 166. 1  
6 174. 1  
7 179. 1  
8 191. 1
```

# SHETH L.U.J. AND SIR M.V. COLLEGE

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for data manipulation and printing. The code includes comments in Hindi and R commands like `open_prop <- prop.table(open_counts)`, `print("Proportion Table: open Prices")`, and `print(open_prop)`. It also includes a large block of data for `open_prop` and similar commands for `close_prop`.
- Console:** Shows the output of the R code, including the printed proportion table for open prices and the execution of `print(open_prop)`.
- Environment Pane:** Lists the objects in the R environment, including `close_prop`, `courses_counts`, `courses_freq`, `courses_prop`, `cross_tab`, `marks_counts`, `marks_freq`, `marks_group`, `marks_prop`, `open_counts`, `open_prop`, `prep_counts`, `profit_freq`, and `profit_group`. Each object is shown with its type and dimensions.
- Files Pane:** Displays a list of files in the current project, including `employee_salary_dataset.csv`, `exams.csv`, `flipkart_e-commerce_sample.csv`, `superstore.csv`, and `random_stock_market_dataset.csv`.