PRACTICAL NO :- 10
AIM :- Creating new variables using transformations and calculations in R. import dataset.

```r
library(dplyr)
library(tidyr)

df <- read.csv("C:/Users/itlab/Downloads/S100/Retail Product.csv", na.strings = c("", "NA"))

df_clean <- df %>%
  mutate(
    Price = replace_na(Price, 0),
    Discount = replace_na(Discount, 0),
    Rating = replace_na(Rating, 0)
  )
print("--- Cleaned Baseline Data ---")
print(head(df_clean))

# 2. METHOD A: ARITHMETIC CALCULATIONS
df_calc <- df_clean %>%
  mutate(
    Discount_Amount = Price * (Discount / 100),  # Step 1: Calc amount off
    Final_Price = Price - Discount_Amount       # Step 2: Subtract from total
  )
print("--- Method A: Arithmetic Results (Final Price) ---")
print(df_calc %>% select(Price, Discount, Final_Price))


# 3. METHOD B: CONDITIONAL LOGIC (ifelse)
df_logic <- df_clean %>%
  mutate(
    Quality_Label = ifelse(Rating > 4.0, "Top Rated", "Average"),
    # Let's add a second logic: Is it expensive?
    Price_Category = ifelse(Price > 4000, "Premium", "Budget")
  )
print("--- Method B: Logic Results (Labels) ---")
print(df_logic %>% select(Rating, Quality_Label, Price, Price_Category))


# 4. METHOD C: TEXT TRANSFORMATION
df_text <- df_clean %>%
  mutate(
    # paste0 connects strings with no separator by default
    # paste connects strings with a space by default
    Product_Summary = paste(Category, "item is", Stock, "at $", Price)
```

NANDINI PANDIT S100
DATA ANALYSIS WITH SAS/SPSS/R.

```
  )
print("--- Method C: Text Transformation ---")
print(head(df_text$Product_Summary))



# 5. ALL TOGETHER (The Standard Workflow)
final_dataset <- df_clean %>%
  mutate(
    Final_Price = Price - (Price * Discount / 100),
    Is_High_Value = ifelse(Final_Price > 2000, TRUE, FALSE),
    Status_Report = paste0("Rating: ", round(Rating, 1), " / Dis: ", Discount, "%")
  )
print("--- Final Combined Dataset ---")
print(head(final_dataset))
```
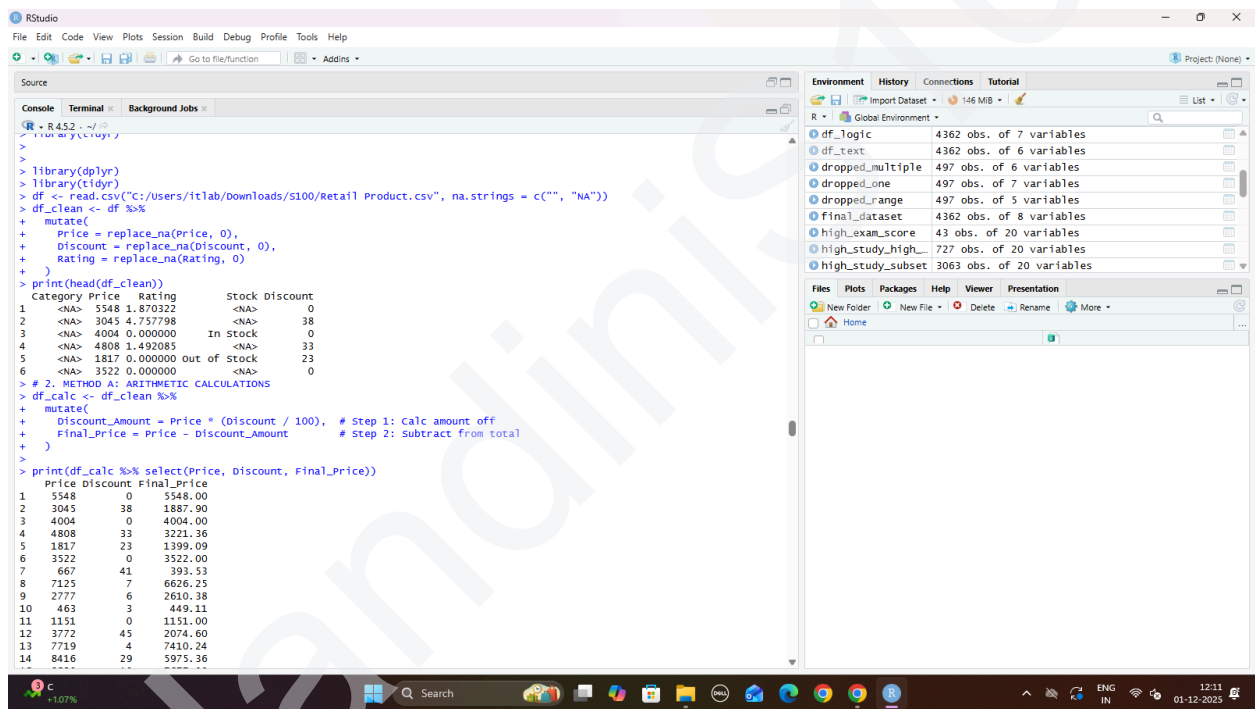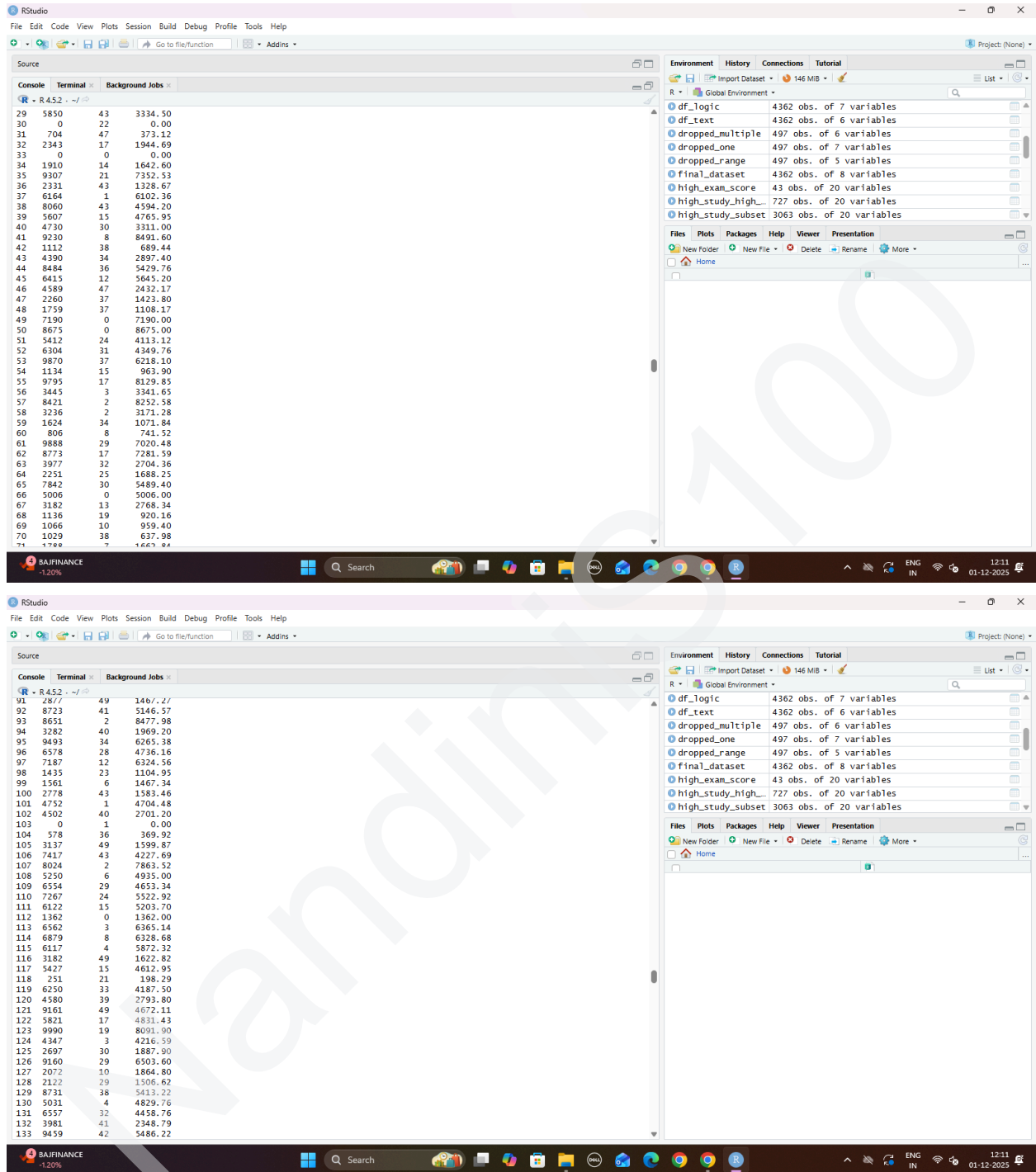

NANDINI PANDIT S100

DATA ANALYSIS WITH SAS/SPSS/R.

MVLU COLLEGE

First console screenshot (R 4.5.2):

29	5850	43	3334.50
30	0	22	0.00
31	704	47	373.12
32	2343	17	1944.69
33	0	0	0.00
34	1910	14	1642.60
35	9307	21	7352.53
36	2331	43	1328.67
37	6164	1	6102.36
38	8060	43	4594.20
39	5607	15	4765.95
40	4730	30	3311.00
41	9230	8	8491.60
42	1112	38	689.44
43	4390	34	2897.40
44	8484	36	5429.76
45	6415	12	5645.20
46	4589	47	2432.17
47	2260	37	1423.80
48	1759	37	1108.17
49	7190	0	7190.00
50	8675	0	8675.00
51	5412	24	4113.12
52	6304	31	4349.76
53	9870	37	6218.10
54	1134	15	963.90
55	9795	17	8129.85
56	3445	3	3341.65
57	8421	2	8252.58
58	3236	2	3171.28
59	1624	34	1071.84
60	806	8	741.52
61	9888	29	7020.48
62	8773	17	7281.59
63	3977	32	2704.36
64	2251	25	1688.25
65	7842	30	5489.40
66	5006	0	5006.00
67	3182	13	2768.34
68	1136	19	920.16
69	1066	10	959.40
70	1029	38	637.98
71	1788	7	1662.84

Environment pane:

- df_logic — 4362 obs. of 7 variables
- df_text — 4362 obs. of 6 variables
- dropped_multiple — 497 obs. of 6 variables
- dropped_one — 497 obs. of 7 variables
- dropped_range — 497 obs. of 5 variables
- final_dataset — 4362 obs. of 8 variables
- high_exam_score — 43 obs. of 20 variables
- high_study_high_... — 727 obs. of 20 variables
- high_study_subset — 3063 obs. of 20 variables

Second console screenshot (R 4.5.2):

91	2877	49	1467.27
92	8723	41	5146.57
93	8651	2	8477.98
94	3282	40	1969.20
95	9493	34	6265.38
96	6578	28	4736.16
97	7187	12	6324.56
98	1435	23	1104.95
99	1561	6	1467.34
100	2778	43	1583.46
101	4752	1	4704.48
102	4502	40	2701.20
103	0	1	0.00
104	578	36	369.92
105	3137	49	1599.87
106	7417	43	4227.69
107	8024	2	7863.52
108	5250	6	4935.00
109	6554	29	4653.34
110	7267	24	5522.92
111	6122	15	5203.70
112	1362	0	1362.00
113	6562	3	6365.14
114	6879	8	6328.68
115	6117	4	5872.32
116	3182	49	1622.82
117	5427	15	4612.95
118	251	21	198.29
119	6250	33	4187.50
120	4580	39	2793.80
121	9161	49	4672.11
122	5821	17	4831.43
123	9990	19	8091.90
124	4347	3	4216.59
125	2697	30	1887.90
126	9160	29	6503.60
127	2072	10	1864.80
128	2122	29	1506.62
129	8731	38	5413.22
130	5031	4	4829.76
131	6557	32	4458.76
132	3981	41	2348.79
133	9459	42	5486.22

Environment pane:

- df_logic — 4362 obs. of 7 variables
- df_text — 4362 obs. of 6 variables
- dropped_multiple — 497 obs. of 6 variables
- dropped_one — 497 obs. of 7 variables
- dropped_range — 497 obs. of 5 variables
- final_dataset — 4362 obs. of 8 variables
- high_exam_score — 43 obs. of 20 variables
- high_study_high_... — 727 obs. of 20 variables
- high_study_subset — 3063 obs. of 20 variables

```
329  6869      49      3503.19
330  3601      22      2808.78
331  6657      47      3528.21
332     0      35         0.00
333  5146      10      4631.40
[ reached 'max' / getOption("max.print") -- omitted 4029 rows ]
> # 3. METHOD B: CONDITIONAL LOGIC (ifelse)
> df_logic <- df_clean %>%
+   mutate(
+     Quality_Label = ifelse(Rating > 4.0, "Top Rated", "Average"),
+     # Let's add a second logic: Is it expensive?
+     Price_Category = ifelse(Price > 4000, "Premium", "Budget")
+   )
> print(df_logic %>% select(Rating, Quality_Label, Price, Price_Category))
    Rating Quality_Label Price Price_Category
1  1.870322       Average  5548        Premium
2  4.757798     Top Rated  3045         Budget
3  0.000000       Average  4004        Premium
4  1.492085       Average  4808        Premium
5  0.000000       Average  1817         Budget
6  0.000000       Average  3522         Budget
7  3.668341       Average   667         Budget
8  4.983998     Top Rated  7125        Premium
9  2.678384       Average  2777         Budget
10 4.626187     Top Rated   463         Budget
11 2.947838       Average  1151         Budget
12 4.890750     Top Rated  3772         Budget
13 2.982242       Average  7719        Premium
14 1.270943       Average  8416        Premium
15 0.000000       Average  8530        Premium
16 3.032832       Average  7936        Premium
17 3.479064       Average  9319        Premium
18 4.097464     Top Rated     0         Budget
19 0.000000       Average  2066         Budget
20 0.000000       Average  1784         Budget
21 0.000000       Average  5589        Premium
22 0.000000       Average     0         Budget
23 2.264051       Average  4176        Premium
24 2.774270       Average   407         Budget
25 0.000000       Average  7011        Premium
26 0.000000       Average  8418        Premium
27 0.000000       Average  3405         Budget
```



```
192 4.340548     Top Rated  4620        Premium
193 1.076074       Average  9424        Premium
194 1.696636       Average  8409        Premium
195 3.624531       Average  9063        Premium
196 1.218936       Average  8405        Premium
197 4.220246     Top Rated  8070        Premium
198 2.576793       Average  7564        Premium
199 2.381393       Average  8555        Premium
200 0.000000       Average  1870         Budget
201 2.893483       Average  6864        Premium
202 2.364554       Average   439         Budget
203 1.803760       Average  1425         Budget
204 3.032992       Average  1612         Budget
205 2.329488       Average  1894         Budget
206 4.469736     Top Rated  8975        Premium
207 0.000000       Average   807         Budget
208 0.000000       Average  2297         Budget
209 0.000000       Average  7800        Premium
210 3.383624       Average  1493         Budget
211 0.000000       Average  2602         Budget
212 4.767811     Top Rated  5381        Premium
213 1.746250       Average  5429        Premium
214 0.000000       Average   734         Budget
215 4.175149     Top Rated  3365         Budget
216 3.319299       Average  9318        Premium
217 0.000000       Average  5220        Premium
218 3.093285       Average   381         Budget
219 0.000000       Average  8707        Premium
220 0.000000       Average  3604         Budget
221 4.391163     Top Rated  5078        Premium
222 0.000000       Average  8070        Premium
223 0.000000       Average   721         Budget
224 0.000000       Average  6921        Premium
225 0.000000       Average  2622         Budget
226 0.000000       Average  7231        Premium
227 0.000000       Average   340         Budget
228 0.000000       Average  9212        Premium
229 0.000000       Average  6472        Premium
230 0.000000       Average  3978         Budget
231 0.000000       Average  5746        Premium
232 3.455945       Average  3956         Budget
233 0.000000       Average  8757        Premium
234 3.088924       Average  2820         Budget
```

NANDINI PANDIT S100
DATA ANALYSIS WITH SAS/SPSS/R.