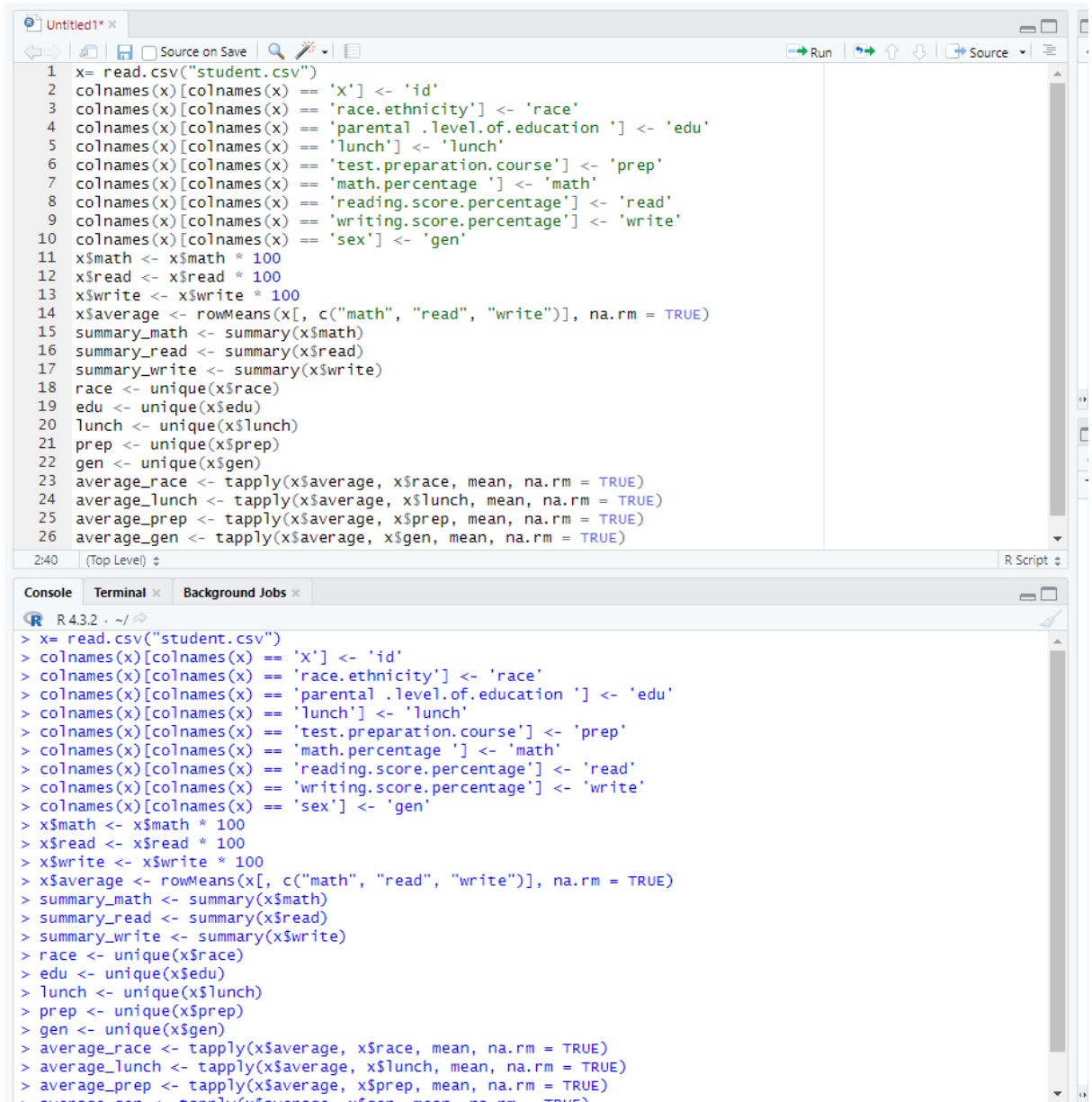


SCS2211 - LABORATORY II

Practical – 02

Activity 1



```
1 x= read.csv("student.csv")
2 colnames(x)[colnames(x) == 'x'] <- 'id'
3 colnames(x)[colnames(x) == 'race.ethnicity'] <- 'race'
4 colnames(x)[colnames(x) == 'parental.level.of.education '] <- 'edu'
5 colnames(x)[colnames(x) == 'lunch'] <- 'lunch'
6 colnames(x)[colnames(x) == 'test.preparation.course'] <- 'prep'
7 colnames(x)[colnames(x) == 'math.percentage '] <- 'math'
8 colnames(x)[colnames(x) == 'reading.score.percentage'] <- 'read'
9 colnames(x)[colnames(x) == 'writing.score.percentage'] <- 'write'
10 colnames(x)[colnames(x) == 'sex'] <- 'gen'
11 x$math <- x$math * 100
12 x$read <- x$read * 100
13 x$write <- x$write * 100
14 x$average <- rowMeans(x[, c("math", "read", "write")], na.rm = TRUE)
15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
```

The screenshot shows the R Studio interface. The top pane displays the R script with 26 lines of code. The code reads a CSV file named 'student.csv', renames columns to 'id', 'race', 'edu', 'lunch', 'prep', 'math', 'read', 'write', and 'gen'. It then scales the 'math', 'read', and 'write' columns by 100. Next, it calculates row means for these three scaled columns, storing the result in 'x\$average'. Finally, it generates summary statistics for each variable and calculates the mean for each category of the categorical variables (race, lunch, prep, gen) using the 'tapply' function. The bottom pane shows the console output, which mirrors the code entered in the script pane.

4.

```

3 colnames(x)[colnames(x) == 'race.ethnicity'] <- 'race'
4 colnames(x)[colnames(x) == 'parental.level.of.education'] <- 'edu'
5 colnames(x)[colnames(x) == 'lunch'] <- 'lunch'
6 colnames(x)[colnames(x) == 'test.preparation.course'] <- 'prep'
7 colnames(x)[colnames(x) == 'math.percentage'] <- 'math'
8 colnames(x)[colnames(x) == 'reading.score.percentage'] <- 'read'
9 colnames(x)[colnames(x) == 'writing.score.percentage'] <- 'write'
10 colnames(x)[colnames(x) == 'sex'] <- 'gen'
11 x$math <- x$math * 100
12 x$read <- x$read * 100
13 x$write <- x$write * 100
14 x$average <- rowMeans(x[, c("math", "read", "write")], na.rm = TRUE)
15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
27 x$math
28

```

28:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 ~ /

```

> x$math
[1] 72 69 90 47 76 71 88 40 64 38 58 40 65 78 50 69 88 18 46 54 66 65 44 69 74
[26] 73 69 67 70 62 69 63 56 40 97 81 74 50 75 57 55 58 53 59 50 65 55 66 57 82
[51] 53 77 53 88 71 33 82 52 58 0 79 39 62 69 59 67 45 60 61 39 58 63 41 61 49
[76] 44 30 80 61 62 47 49 50 72 42 73 76 71 58 73 65 27 71 43 79 78 65 63 58 65
[101] 79 68 85 60 98 58 87 66 52 70 77 62 54 51 99 84 75 78 51 55 79 91 88 63 83
[126] 87 72 65 82 51 89 53 87 75 74 58 51 70 59 71 76 59 42 57 88 22 88 73 68 100
[151] 62 77 59 54 62 70 66 60 61 66 82 75 49 52 81 96 53 58 68 67 72 94 79 63 43
[176] 81 46 71 52 97 62 46 50 65 45 65 80 62 48 77 66 76 62 77 69 61 59 55 45 78
[201] 67 65 69 57 59 74 82 81 74 58 80 35 42 60 87 84 83 34 66 61 56 87 55 86 52
[226] 45 72 57 68 88 76 46 67 92 83 80 63 64 54 84 73 80 56 59 75 85 89 58 65 68
[251] 47 71 60 80 54 62 64 78 70 65 64 79 44 99 76 59 63 69 88 71 69 58 47 65 88
[276] 83 85 59 65 73 53 45 73 70 37 81 97 67 88 77 76 86 63 65 78 67 46 71 40 90
[301] 81 56 67 80 74 69 99 51 53 49 73 66 67 68 59 71 77 83 63 56 67 75 71 43 41
[326] 82 61 28 82 41 71 47 62 90 83 61 76 49 24 35 58 61 69 67 79 72 62 77 75 87
[351] 52 66 63 46 59 61 63 42 59 80 58 85 52 27 59 49 69 61 44 73 84 45 74 82 59
[376] 46 80 85 71 66 80 87 79 38 38 67 64 57 62 73 73 77 76 57 65 48 50 85 74 60
[401] 59 53 49 88 54 63 65 82 52 87 70 84 71 63 51 84 71 74 68 57 82 57 47 59 41
[426] 62 86 69 65 68 64 61 61 47 73 50 75 75 70 89 67 78 59 73 79 67 69 86 47 81
[451] 64 100 65 65 53 37 79 53 100 72 53 54 71 77 75 84 26 72 77 91 83 63 68 59 90
[476] 71 76 80 55 76 73 52 68 59 49 70 61 60 64 79 65 64 83 81 54 68 54 59 66 76
[501] 74 94 63 95 40 82 68 55 79 86 76 64 62 54 77 76 74 66 66 67 71 91 69 54 53
[526] 68 56 36 29 62 68 47 62 79 73 66 51 51 85 97 75 79 81 82 64 78 92 72 62 79
[551] 79 87 40 77 53 32 55 61 53 73 74 63 96 63 48 48 92 61 63 68 71 91 53 50 74
[576] 40 61 81 48 53 81 77 63 73 69 65 55 44 54 48 58 71 68 74 92 56 30 53 69 65

```

5.

```

4 colnames(x)[colnames(x) == 'parental.level.of.education '] <- 'edu'
5 colnames(x)[colnames(x) == 'lunch'] <- 'lunch'
6 colnames(x)[colnames(x) == 'test.preparation.course'] <- 'prep'
7 colnames(x)[colnames(x) == 'math.percentage '] <- 'math'
8 colnames(x)[colnames(x) == 'reading.score.percentage'] <- 'read'
9 colnames(x)[colnames(x) == 'writing.score.percentage'] <- 'write'
10 colnames(x)[colnames(x) == 'sex'] <- 'gen'
11 x$math <- x$math * 100
12 x$read <- x$read * 100
13 x$write <- x$write * 100
14 x$average <- rowMeans(x[, c("math", "read", "write")], na.rm = TRUE)
15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
27 x$math
28 x$average
29

```

29:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 ~ /

```

> x$average
[1] 72.66667 82.33333 92.66667 49.33333 76.33333 77.33333 91.66667 40.66667 65.00000 49.33333
[11] 54.66667 45.00000 73.00000 73.33333 53.66667 74.00000 87.66667 26.00000 44.66667 57.66667
[21] 66.00000 70.00000 50.33333 71.66667 75.00000 73.00000 59.33333 70.33333 68.33333 69.00000
[31] 72.33333 63.00000 64.33333 40.00000 88.66667 80.33333 79.33333 57.66667 84.33333 56.66667
[41] 56.66667 66.33333 58.66667 63.33333 53.33333 58.66667 60.66667 71.00000 69.00000 82.66667
[51] 52.00000 71.33333 46.33333 80.33333 80.66667 39.00000 84.33333 52.00000 58.33333 9.00000
[61] 75.00000 37.33333 59.33333 73.33333 58.66667 64.00000 39.66667 68.66667 58.33333 53.33333
[71] 64.66667 60.33333 46.66667 58.00000 46.33333 41.00000 26.00000 79.66667 69.00000 66.00000
[81] 48.66667 46.33333 50.33333 66.33333 38.33333 78.33333 82.33333 72.00000 65.00000 80.33333
[91] 70.33333 32.33333 73.66667 46.00000 85.66667 80.33333 64.33333 68.33333 62.33333 64.66667
[101] 71.00000 72.00000 88.33333 50.33333 91.33333 65.66667 95.66667 64.33333 66.00000 68.66667
[111] 88.00000 55.33333 51.33333 54.33333 99.66667 78.33333 80.66667 79.66667 58.33333 63.00000
[121] 86.66667 90.66667 91.33333 58.66667 78.66667 89.33333 69.00000 72.00000 79.33333 50.33333
[131] 85.00000 43.33333 77.00000 80.00000 76.00000 53.66667 48.66667 60.33333 64.00000 67.00000
[141] 73.00000 61.66667 50.33333 49.00000 79.66667 31.33333 82.33333 69.00000 74.66667 97.66667
[151] 66.00000 70.66667 65.00000 50.00000 63.33333 82.33333 72.66667 60.00000 78.00000 64.00000
[161] 78.00000 82.66667 51.33333 51.33333 88.00000 98.66667 51.66667 70.66667 76.33333 70.66667
[171] 73.00000 86.66667 82.00000 66.66667 49.33333 86.33333 52.66667 75.00000 61.33333 99.00000
[181] 68.33333 58.66667 49.00000 68.00000 48.66667 65.66667 73.66667 65.33333 48.33333 84.00000
[191] 66.66667 79.33333 64.00000 67.00000 79.33333 56.00000 64.33333 54.00000 51.00000 77.66667
[201] 79.00000 74.33333 71.66667 64.66667 47.33333 74.33333 68.66667 79.00000 77.00000 61.66667
[211] 79.33333 30.00000 54.66667 55.66667 86.33333 80.66667 85.66667 38.33333 71.00000 57.66667
[221] 66.00000 81.66667 60.66667 80.33333 63.66667 51.66667 71.66667 53.66667 68.00000 92.33333
[231] 68.66667 43.66667 78.66667 85.66667 83.00000 77.33333 65.33333 66.00000 52.33333 81.33333
[241] 60.00000 82.00000 54.33333 51.00000 72.66667 72.33333 70.66667 65.33333 62.66667 60.33333

```

6.

```

Untitled1 * x
Source on Save
Run
Source

6 colnames(x)[colnames(x) == 'test.preparation.course'] <- 'prep'
7 colnames(x)[colnames(x) == 'math.percentage '] <- 'math'
8 colnames(x)[colnames(x) == 'reading.score.percentage'] <- 'read'
9 colnames(x)[colnames(x) == 'writing.score.percentage'] <- 'write'
10 colnames(x)[colnames(x) == 'sex'] <- 'gen'
11 x$math <- x$math * 100
12 x$read <- x$read * 100
13 x$write <- x$write * 100
14 x$average <- rowMeans(x[, c("math", "read", "write")], na.rm = TRUE)
15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
27 x$math
28 x$average
29 summary_math
30 summary_read
31 summary_write

31:14 (Top Level) R Script

Console Terminal Background Jobs
R 4.3.2 . ~/
[851] 68.33333 64.00000 88.33333 70.00000 60.33333 96.66667 69.33333 75.00000 49.00000 77.33333
[861] 56.00000 84.66667 39.66667 74.00000 93.66667 84.00000 54.66667 48.00000 74.66667 50.33333
[871] 54.66667 72.66667 81.33333 87.33333 55.33333 70.00000 82.33333 57.33333 70.33333 71.33333
[881] 64.33333 70.33333 69.33333 48.33333 52.00000 72.66667 96.00000 61.66667 69.33333 47.66667
[891] 87.33333 87.33333 63.33333 79.00000 63.33333 34.66667 30.66667 73.33333 61.33333 75.00000
[901] 86.00000 75.00000 41.00000 97.66667 78.33333 80.66667 52.00000 84.66667 71.33333 68.00000
[911] 46.66667 78.33333 59.66667 58.33333 54.00000 68.00000 100.00000 52.33333 71.33333 92.66667
[921] 68.66667 44.66667 68.00000 61.33333 71.00000 66.00000 57.33333 65.66667 44.33333 51.66667
[931] 70.33333 60.33333 68.33333 73.00000 91.66667 63.66667 59.00000 57.33333 83.66667 71.33333
[941] 70.00000 88.33333 70.33333 61.33333 62.33333 57.66667 81.33333 55.66667 50.33333 68.33333
[951] 79.33333 78.33333 73.33333 54.66667 66.66667 63.66667 84.00000 97.33333 55.66667 75.66667
[961] 59.33333 51.66667 100.00000 72.00000 62.33333 68.00000 66.00000 62.00000 70.33333 79.66667
[971] 96.33333 73.00000 54.33333 58.33333 61.33333 74.33333 60.66667 61.66667 48.00000 93.33333
[981] 18.33333 79.00000 83.33333 85.33333 77.00000 54.00000 50.00000 77.33333 44.66667 78.66667
[991] 80.66667 75.00000 69.00000 69.33333 62.66667 94.00000 57.33333 65.00000 74.33333 83.00000
> summary_math
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00   57.00   66.00   66.09   77.00   100.00
> summary_read
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 17.00   59.00   70.00   69.17   79.00   100.00
> summary_write
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 10.00   57.75   69.00   68.05   79.00   100.00
>

```

7.

```

11 x$math <- x$math * 100
12 x$read <- x$read * 100
13 x$write <- x$write * 100
14 x$average <- rowMeans(x[, c("math", "read", "write")], na.rm = TRUE)
15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
27 x$math
28 x$average
29 summary_math
30 summary_read
31 summary_write
32 race
33 edu
34 lunch
35 prep
36 gen

```

32:5 (Top Level) R Script

Console Terminal Background Jobs

```

R 4.3.2 ~ /
[951] 79.33333 78.33333 73.33333 54.66667 66.66667 63.66667 84.00000 97.33333 55.66667 75.66667
[961] 59.33333 51.66667 100.00000 72.00000 62.33333 68.00000 66.00000 62.00000 70.33333 79.66667
[971] 96.33333 73.00000 54.33333 58.33333 61.33333 74.33333 60.66667 61.66667 48.00000 93.33333
[981] 18.33333 79.00000 83.33333 85.33333 77.00000 54.00000 50.00000 77.33333 44.66667 78.66667
[991] 80.66667 75.00000 69.00000 69.33333 62.66667 94.00000 57.33333 65.00000 74.33333 83.00000
> summary_math
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00  57.00   66.00   66.09  77.00  100.00
> summary_read
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 17.00  59.00   70.00  69.17  79.00  100.00
> summary_write
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 10.00  57.75   69.00  68.05  79.00  100.00
> race
[1] "group B" "group C" "group A" "group D" "group E"
> edu
NULL
> lunch
[1] "standard" "free/reduced"
> prep
[1] "none" "completed"
> gen
[1] "F" "M"

```

8.

The screenshot shows the R Studio environment. The top pane contains R code for summarizing and averaging data across different groups. The bottom pane shows the console output for these operations.

```

15 summary_math <- summary(x$math)
16 summary_read <- summary(x$read)
17 summary_write <- summary(x$write)
18 race <- unique(x$race)
19 edu <- unique(x$edu)
20 lunch <- unique(x$lunch)
21 prep <- unique(x$prep)
22 gen <- unique(x$gen)
23 average_race <- tapply(x$average, x$race, mean, na.rm = TRUE)
24 average_lunch <- tapply(x$average, x$lunch, mean, na.rm = TRUE)
25 average_prep <- tapply(x$average, x$prep, mean, na.rm = TRUE)
26 average_gen <- tapply(x$average, x$gen, mean, na.rm = TRUE)
27 x$math
28 x$average
29 summary_math
30 summary_read
31 summary_write
32 race
33 edu
34 lunch
35 prep
36 gen
37 average_race
38 average_lunch
39 average_prep
40 average_gen

```

Console Output:

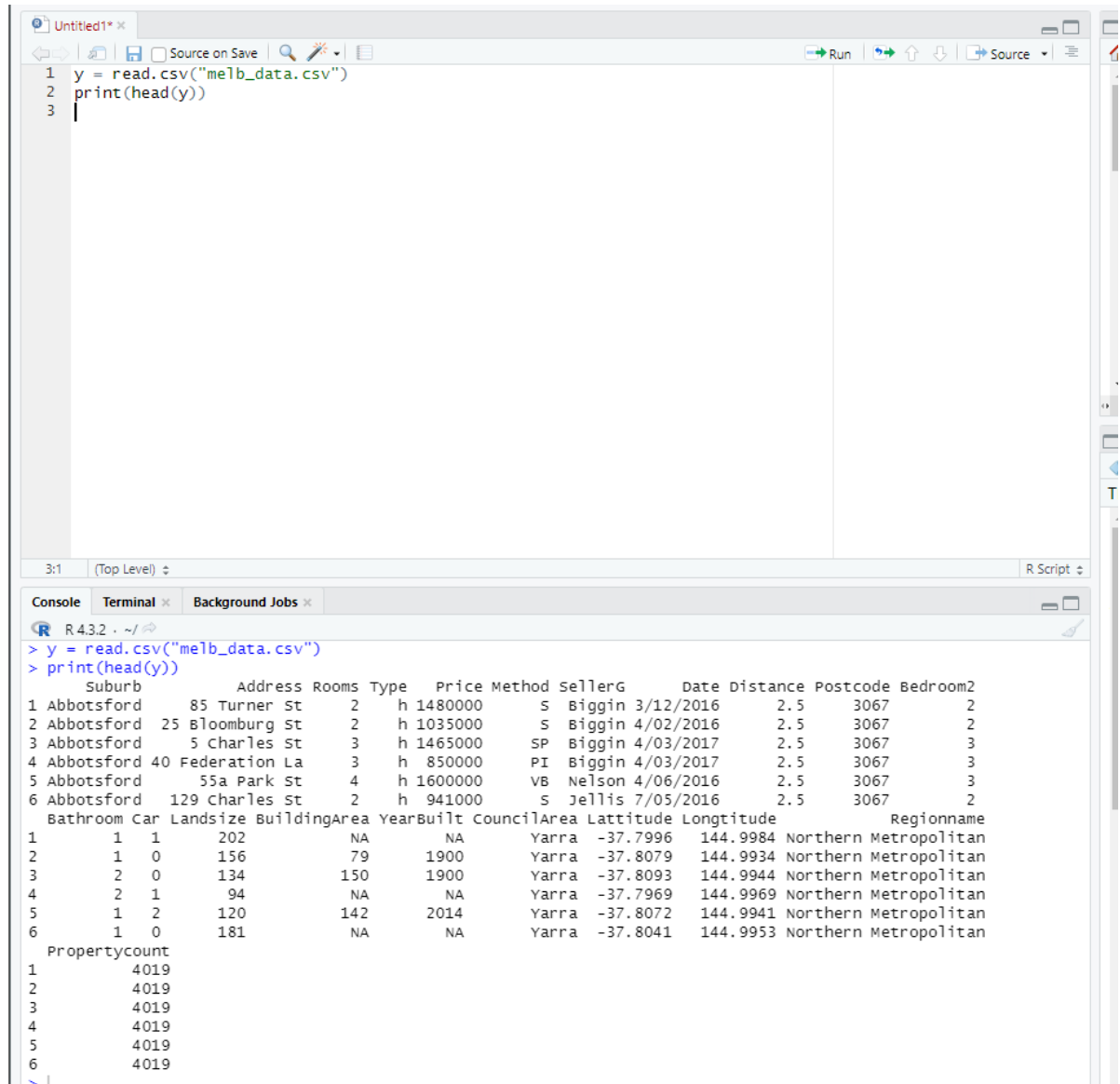
```

R 4.3.2 ~ /
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 10.00  57.75   69.00   68.05   79.00  100.00
> race
[1] "group B" "group C" "group A" "group D" "group E"
> edu
NULL
> lunch
[1] "standard"      "free/reduced"
> prep
[1] "none"          "completed"
> gen
[1] "F" "M"
> average_race
  group A  group B  group C  group D  group E
62.99251 65.46842 67.13166 69.17939 72.75238
> average_lunch
free/reduced    standard
62.19906       70.83721
> average_prep
completed    none
72.66946    65.03894
> average_gen
      F      M
69.56950 65.83748
>

```

Activity 2

3.



The screenshot shows the R Studio environment. The script editor at the top contains the following code:

```
1 y = read.csv("melb_data.csv")
2 print(head(y))
3
```

The console at the bottom displays the output of the code. It shows the first six rows of the 'melb_data.csv' file, followed by the first six rows of the 'Propertycount' dataset.

```
> y = read.csv("melb_data.csv")
> print(head(y))
  Suburb      Address Rooms Type   Price Method SellerG      Date Distance Postcode Bedroom2
1 Abbotsford 85 Turner St    2   h 1480000    S Biggin 3/12/2016      2.5    3067        2
2 Abbotsford 25 Bloomburg St  2   h 1035000    S Biggin 4/02/2016      2.5    3067        2
3 Abbotsford 5 Charles St   3   h 1465000    SP Biggin 4/03/2017      2.5    3067        3
4 Abbotsford 40 Federation La 3   h 850000    PI Biggin 4/03/2017      2.5    3067        3
5 Abbotsford 55a Park St    4   h 1600000    VB Nelson 4/06/2016      2.5    3067        3
6 Abbotsford 129 Charles St  2   h 941000    S Jellis 7/05/2016      2.5    3067        2
  Bathroom Car Landsize BuildingArea YearBuilt CouncilArea Latitude Longitude Regionname
1      1      1      202          NA          NA      Yarra  -37.7996    144.9984 Northern Metropolitan
2      1      0      156          79      1900      Yarra  -37.8079    144.9934 Northern Metropolitan
3      2      0      134          150      1900      Yarra  -37.8093    144.9944 Northern Metropolitan
4      2      1       94          NA          NA      Yarra  -37.7969    144.9969 Northern Metropolitan
5      1      2      120          142      2014      Yarra  -37.8072    144.9941 Northern Metropolitan
6      1      0      181          NA          NA      Yarra  -37.8041    144.9953 Northern Metropolitan
  Propertycount
1      4019
2      4019
3      4019
4      4019
5      4019
6      4019
>
```

4.

The screenshot shows an R Studio window with a script editor and a console. The script editor contains the following code:

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colsums(is.na(y))
4 missing_values
5

```

The console output shows the execution of the script. It starts with the R version and session information, followed by the output of the `print(head(y))` command, which displays the first six rows of the `melb_data.csv` file. The output is as follows:

```

5 Abbotsford 55a Park St 4 h 1600000 VB Nelson 4/06/2016 2.5 3067 3
6 Abbotsford 129 Charles St 2 h 941000 S Jellis 7/05/2016 2.5 3067 2
  Bathroom Car Landsize BuildingArea YearBuilt CouncilArea Latitude Longitude Regionname
1 1 1 202 NA NA Yarra -37.7996 144.9984 Northern Metropolitan
2 1 0 156 79 1900 Yarra -37.8079 144.9934 Northern Metropolitan
3 2 0 134 150 1900 Yarra -37.8093 144.9944 Northern Metropolitan
4 2 1 94 NA NA Yarra -37.7969 144.9969 Northern Metropolitan
5 1 2 120 142 2014 Yarra -37.8072 144.9941 Northern Metropolitan
6 1 0 181 NA NA Yarra -37.8041 144.9953 Northern Metropolitan

```

Below the head output, the `missing_values` object is displayed, showing the number of missing values for each column:

```

Propertycount
1 4019
2 4019
3 4019
4 4019
5 4019
6 4019
> missing_values <- colsums(is.na(y))
> missing_values
  Suburb Address Rooms Type Price Method SellerG
0 0 0 0 0 0 0 0
  Date Distance Postcode Bedroom2 Bathroom Car Landsize
0 0 0 0 0 0 62 0
BuildingArea YearBuilt CouncilArea Latitude Longitude Regionname Propertycount
6450 5375 0 0 0 0 0 0

```


5.

The screenshot shows the RStudio IDE with a script editor and a console. The script editor contains the following R code:

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colsums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7

```

The console shows the output of the code, including the head of the data frame, the missing values matrix, and the mean year built.

```

R 4.3.2 . ~/
1      1      1      202      NA      NA      Yarra -37.7996 144.9984 Northern Metropolitan
2      1      0      156      79     1900      Yarra -37.8079 144.9934 Northern Metropolitan
3      2      0      134     150     1900      Yarra -37.8093 144.9944 Northern Metropolitan
4      2      1       94      NA      NA      Yarra -37.7969 144.9969 Northern Metropolitan
5      1      2      120     142     2014      Yarra -37.8072 144.9941 Northern Metropolitan
6      1      0      181      NA      NA      Yarra -37.8041 144.9953 Northern Metropolitan
Propertycount
1      4019
2      4019
3      4019
4      4019
5      4019
6      4019
> missing_values <- colsums(is.na(y))
> missing_values
      Suburb      Address      Rooms      Type      Price      Method      SellerG
      0      0      0      0      0      0      0
      Date      Distance      Postcode      Bedroom2      Bathroom      Car      Landsize
      0      0      0      0      0      62      0
BuildingArea      YearBuilt      CouncilArea      Latitude      Longitude      Regionname      Propertycount
6450      5375      0      0      0      0      0
> mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
> mean_year_built
[1] 1964.684
>

```

6.

The screenshot shows the RStudio IDE with a script editor and a console. The script editor contains the following R code:

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colSums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7 emp_sal <- data.frame(
8   Emp_ID = c(11, 12, 13, 14, 15),
9   Dep = c("Sales", "HR", "Sales", "HR", "Sales"),
10  Basic = c(25450, 22500, 21000, 23500, 15000),
11  Allowances = c(5200, 4500, 3100, 2600, 1800)
12 )
13 emp_sal
14 |

```

The console shows the execution of the code, including the output of `colSums(is.na(y))`, the mean of `YearBuilt`, and the creation of the `emp_sal` data frame.

```

> missing_values <- colSums(is.na(y))
> missing_values
      Suburb      Address      Rooms      Type      Price      Method      SellerG
      0         0         0         0         0         0         0
      Date      Distance      Postcode      Bedroom2      Bathroom      Car      Landsize
      0         0         0         0         0         62         0
      BuildingArea      YearBuilt      CouncilArea      Latitude      Longitude      Regionname      Propertycount
      6450         5375         0         0         0         0         0
> mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
> mean_year_built
[1] 1964.684
> emp_sal <- data.frame(
+   Emp_ID = c(11, 12, 13, 14, 15),
+   Dep = c("Sales", "HR", "Sales", "HR", "Sales"),
+   Basic = c(25450, 22500, 21000, 23500, 15000),
+   Allowances = c(5200, 4500, 3100, 2600, 1800)
+ )
> emp_sal
  Emp_ID Dep Basic Allowances
1    11 Sales 25450       5200
2    12  HR 22500       4500
3    13 Sales 21000       3100
4    14  HR 23500       2600
5    15 Sales 15000       1800
>

```

7.

The screenshot shows the RStudio environment. The top pane displays an R script with the following code:

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colSums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7 emp_sal <- data.frame(
8   Emp_ID = c(11, 12, 13, 14, 15),
9   Dep = c("Sales", "HR", "Sales", "HR", "Sales"),
10  Basic = c(25450, 22500, 21000, 23500, 15000),
11  Allowances = c(5200, 4500, 3100, 2600, 1800)
12 )
13 emp_sal
14 emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
15 emp_sal
16 |

```

The bottom pane shows the console output for the executed code:

```

R 4.3.2 ~|
+ Allowances = c(5200, 4500, 3100, 2600, 1800)
+ )
> emp_sal
  Emp_ID Dep Basic Allowances
1    11 Sales 25450        5200
2    12  HR 22500        4500
3    13 Sales 21000        3100
4    14  HR 23500        2600
5    15 Sales 15000        1800
> emp_sal
  Emp_ID Dep Basic Allowances
1    11 Sales 25450        5200
2    12  HR 22500        4500
3    13 Sales 21000        3100
4    14  HR 23500        2600
5    15 Sales 15000        1800
> emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
> emp_sal
  Emp_ID Dep Basic Allowances net_sal
1    11 Sales 25450        5200 30650
2    12  HR 22500        4500 27000
3    13 Sales 21000        3100 24100
4    14  HR 23500        2600 26100
5    15 Sales 15000        1800 16800
> |

```

8.

The screenshot shows the RStudio IDE with a script editor and a console. The script editor contains the following R code:

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colsums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7 emp_sal <- data.frame(
8   Emp_ID = c(11, 12, 13, 14, 15),
9   Dep = c("Sales", "HR", "Sales", "HR", "Sales"),
10  Basic = c(25450, 22500, 21000, 23500, 15000),
11  Allowances = c(5200, 4500, 3100, 2600, 1800)
12 )
13 emp_sal
14 emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
15 emp_sal
16 net_salaries <- emp_sal$net_sal[emp_sal$Dep == "Sales"]
17 net_salaries
18

```

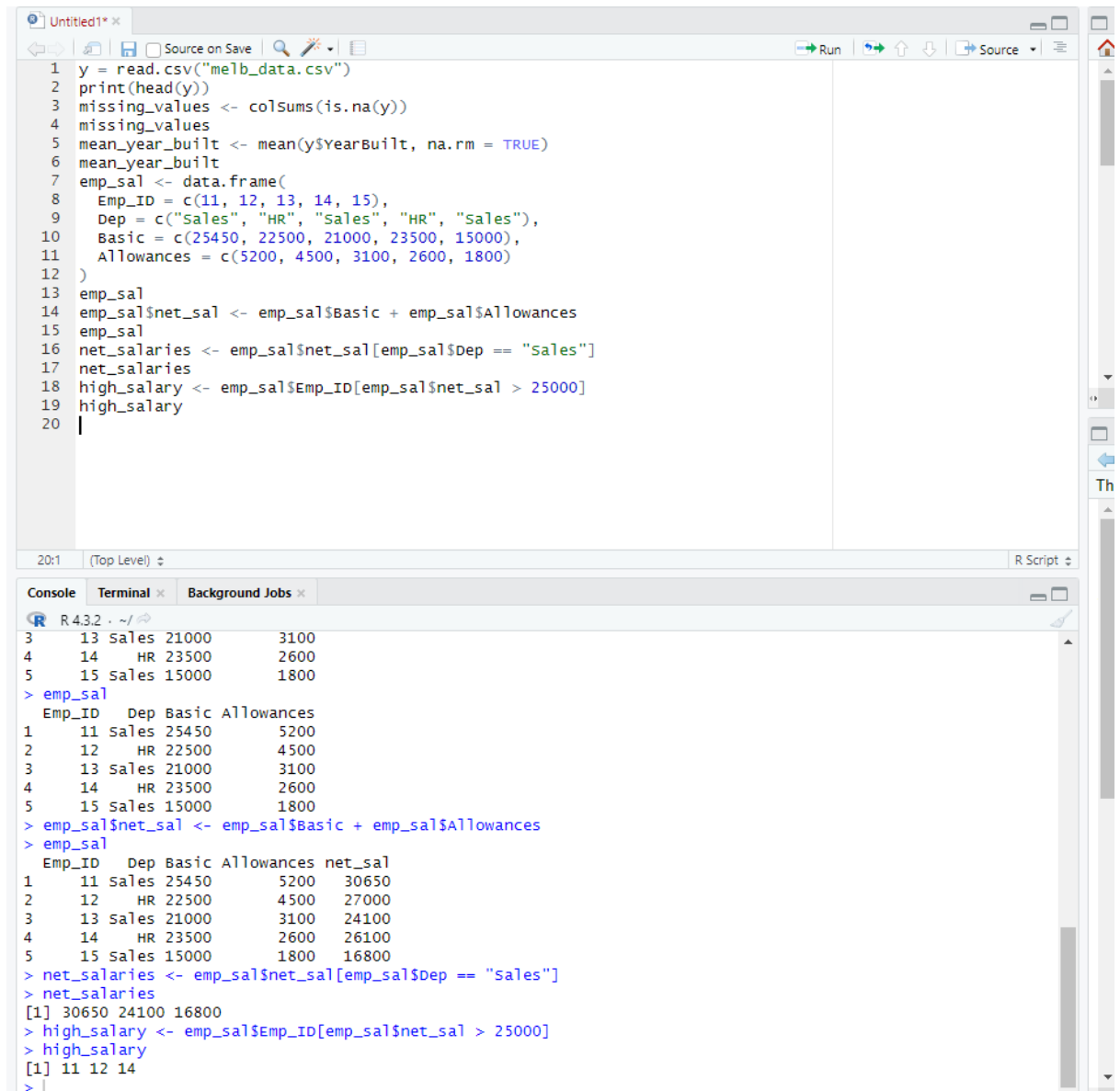
The console shows the output of the script, including the data frame 'emp_sal' and the resulting 'net_salaries' vector.

```

R 4.3.2 . ~/
Emp_ID Dep Basic Allowances
1 11 Sales 25450 5200
2 12 HR 22500 4500
3 13 Sales 21000 3100
4 14 HR 23500 2600
5 15 Sales 15000 1800
> emp_sal
Emp_ID Dep Basic Allowances
1 11 Sales 25450 5200
2 12 HR 22500 4500
3 13 Sales 21000 3100
4 14 HR 23500 2600
5 15 Sales 15000 1800
> emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
> emp_sal
Emp_ID Dep Basic Allowances net_sal
1 11 Sales 25450 5200 30650
2 12 HR 22500 4500 27000
3 13 Sales 21000 3100 24100
4 14 HR 23500 2600 26100
5 15 Sales 15000 1800 16800
> net_salaries <- emp_sal$net_sal[emp_sal$Dep == "Sales"]
> net_salaries
[1] 30650 24100 16800
>

```

9.



The screenshot shows an R Studio window with a script editor and a console. The script editor contains R code for reading a CSV file, calculating mean values, and creating a data frame. The console shows the execution of the script, including the creation of the 'emp_sal' data frame and the calculation of net salaries.

```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colsums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7 emp_sal <- data.frame(
8   Emp_ID = c(11, 12, 13, 14, 15),
9   Dep = c("Sales", "HR", "Sales", "HR", "Sales"),
10  Basic = c(25450, 22500, 21000, 23500, 15000),
11  Allowances = c(5200, 4500, 3100, 2600, 1800)
12 )
13 emp_sal
14 emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
15 emp_sal
16 net_salaries <- emp_sal$net_sal[emp_sal$Dep == "Sales"]
17 net_salaries
18 high_salary <- emp_sal$Emp_ID[emp_sal$net_sal > 25000]
19 high_salary
20 |

```

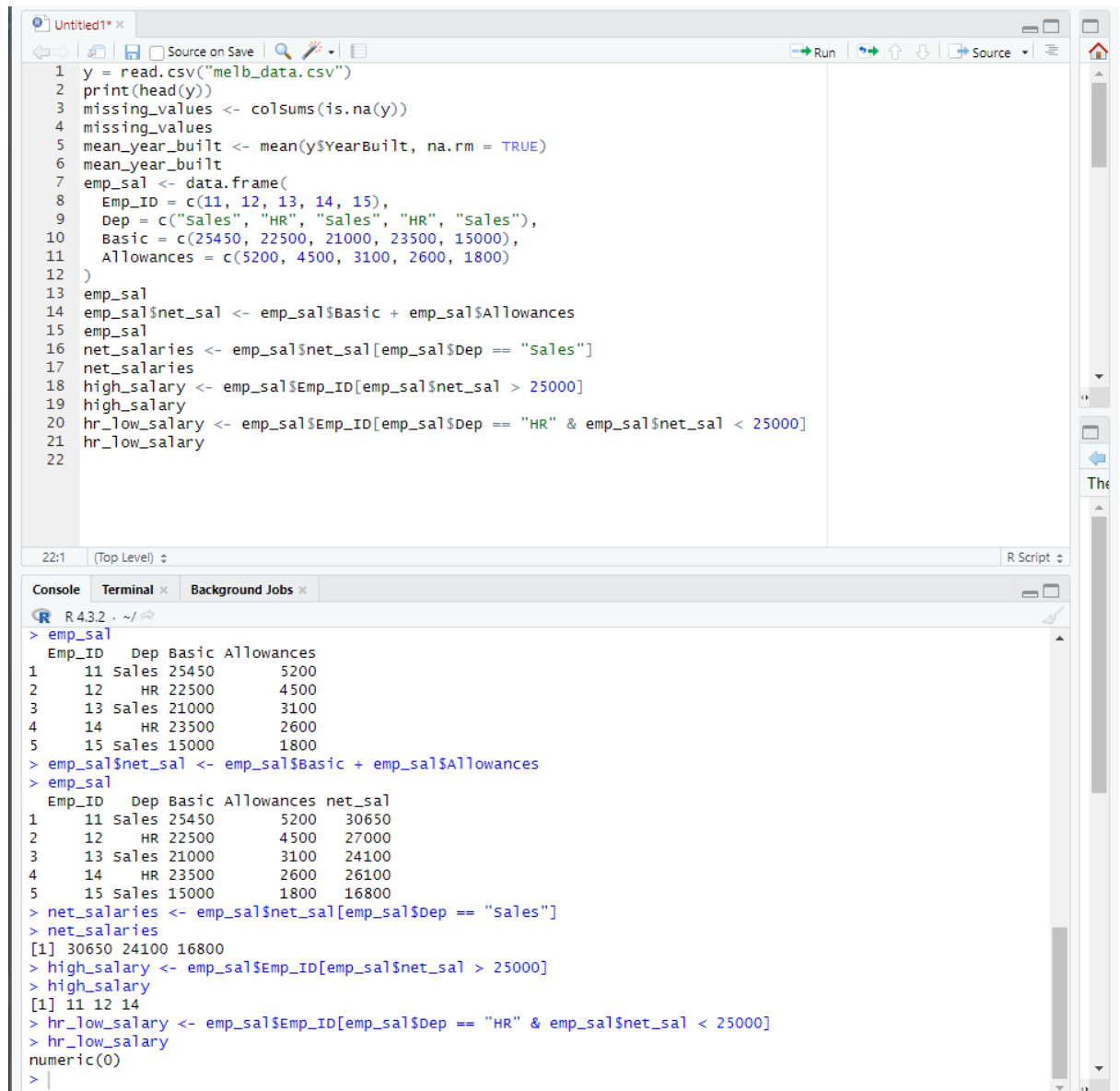
The console output shows the following results:

```

R 4.3.2 ~ /
3 13 Sales 21000 3100
4 14 HR 23500 2600
5 15 Sales 15000 1800
> emp_sal
  Emp_ID Dep Basic Allowances
1 11 Sales 25450 5200
2 12 HR 22500 4500
3 13 Sales 21000 3100
4 14 HR 23500 2600
5 15 Sales 15000 1800
> emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
> emp_sal
  Emp_ID Dep Basic Allowances net_sal
1 11 Sales 25450 5200 30650
2 12 HR 22500 4500 27000
3 13 Sales 21000 3100 24100
4 14 HR 23500 2600 26100
5 15 Sales 15000 1800 16800
> net_salaries <- emp_sal$net_sal[emp_sal$Dep == "Sales"]
> net_salaries
[1] 30650 24100 16800
> high_salary <- emp_sal$Emp_ID[emp_sal$net_sal > 25000]
> high_salary
[1] 11 12 14
> |

```

10.



```

1 y = read.csv("melb_data.csv")
2 print(head(y))
3 missing_values <- colsums(is.na(y))
4 missing_values
5 mean_year_built <- mean(y$YearBuilt, na.rm = TRUE)
6 mean_year_built
7 emp_sal <- data.frame(
8   Emp_ID = c(11, 12, 13, 14, 15),
9   Dep = c("sales", "HR", "sales", "HR", "sales"),
10  Basic = c(25450, 22500, 21000, 23500, 15000),
11  Allowances = c(5200, 4500, 3100, 2600, 1800)
12 )
13 emp_sal
14 emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
15 emp_sal
16 net_salaries <- emp_sal$net_sal[emp_sal$Dep == "sales"]
17 net_salaries
18 high_salary <- emp_sal$Emp_ID[emp_sal$net_sal > 25000]
19 high_salary
20 hr_low_salary <- emp_sal$Emp_ID[emp_sal$Dep == "HR" & emp_sal$net_sal < 25000]
21 hr_low_salary
22

```

Console

```

> emp_sal
  Emp_ID Dep Basic Allowances
1    11 sales 25450      5200
2    12  HR 22500      4500
3    13 sales 21000      3100
4    14  HR 23500      2600
5    15 sales 15000      1800
> emp_sal$net_sal <- emp_sal$Basic + emp_sal$Allowances
> emp_sal
  Emp_ID Dep Basic Allowances net_sal
1    11 sales 25450      5200 30650
2    12  HR 22500      4500 27000
3    13 sales 21000      3100 24100
4    14  HR 23500      2600 26100
5    15 sales 15000      1800 16800
> net_salaries <- emp_sal$net_sal[emp_sal$Dep == "sales"]
> net_salaries
[1] 30650 24100 16800
> high_salary <- emp_sal$Emp_ID[emp_sal$net_sal > 25000]
> high_salary
[1] 11 12 14
> hr_low_salary <- emp_sal$Emp_ID[emp_sal$Dep == "HR" & emp_sal$net_sal < 25000]
> hr_low_salary
numeric(0)
>

```