

## PRACTICAL NO 7

AIM: Selecting and dropping variables using select() in R. import dataset.

```

1 #Z. Selecting and dropping variables using select() in R.
2 #
3 # 1. IMPORT DATASET
4 #
5 library(dplyr) # select() is part of the dplyr package
6
7 # Import the CSV file
8 housing <- read_csv("R/netflix_titles.csv")
9
10 print("---- Original Dataset (First 3 rows) ----")
11 print(head(housing, 3))
12
13 #
14 # 2. SELECTING VARIABLES (Keeping columns)
15 #
16
17 # Method A: Select specific columns by name
18 # Scenario: We only want the Crime rate (crim), Rooms (rm), and value (medv)
19 # (Mapped to Netflix: type, title, release_year)
20 selected_cols <- housing %>%
21   select(type, title, release_year)
22
23 print("---- Selected Specific Columns ----")
24 print(head(selected_cols, 3))
25
26 # Method B: Select a range of adjacent columns
27 # Scenario: Select everything from 'crim' to 'nox'
28 # (Mapped to Netflix: type: title: rating)
29 range_cols <- housing %>%
30   select(type:rating)
31
32 print("---- Selected Range of Columns ----")
33 print(head(range_cols, 3))
34
75:1 (Untitled)

```

```

35 # Method C: Select using helper functions (e.g., starts_with)
36 # Scenario: Select columns that start with 'r' (rm, rad)
37 # (Netflix columns starting with 'r' include: release_year, rating)
38 starts_with_r <- housing %>%
39   select(starts_with("r"))
40
41 print("---- Selected columns starting with 'r' ----")
42 print(head(starts_with_r, 3))
43
44 #
45 # 3. DROPPING VARIABLES (Removing columns)
46 #
47 # We use the minus sign (-) to remove variables
48
49 # Method A: Drop a single specific column
50 # Scenario: Remove the 'chas'
51 # (Mapped to Netflix: drop 'country')
52 dropped_one <- housing %>%
53   select(-country)
54
55 print("---- Dataset with 'country' dropped ----")
56 print(names(dropped_one)) # Printing names to verify it's gone
57
58 # Method B: Drop multiple columns
59 # Scenario: Remove 'zn' and 'indus'
60 # (Mapped to Netflix: drop 'cast' and 'director')
61 dropped_multiple <- housing %>%
62   select(-cast, -director)
63
64 print("---- Dataset with 'cast' and 'director' dropped ----")
65 print(names(dropped_multiple))
66
67 # Method C: Drop a range of columns
68 # Scenario: Remove everything from 'age' to 'tax'
69
75:1 (Untitled)

```

```

42 print(head(starts_with_r, 3))
43
44 # =====
45 # 3. DROPPING VARIABLES (Removing columns)
46 # =====
47 # We use the minus sign (-) to remove variables
48
49 # Method A: Drop a single specific column
50 # Scenario: Remove the 'chas'
51 # (Mapped to Netflix: drop 'country')
52 dropped_one <- housing %>%
53   select(-country)
54
55 print("--- Dataset with 'country' dropped ---")
56 print(names(dropped_one)) # Printing names to verify it's gone
57
58 # Method B: Drop multiple columns
59 # Scenario: Remove 'zn' and 'indus'
60 # (Mapped to Netflix: drop 'cast' and 'director')
61 dropped_multiple <- housing %>%
62   select(-cast, -director)
63
64 print("--- Dataset with 'cast' and 'director' dropped ---")
65 print(names(dropped_multiple))
66
67 # Method C: Drop a range of columns
68 # Scenario: Remove everything from 'age' to 'tax'
69 # (Mapped to Netflix: duration - listed_in)
70 dropped_range <- housing %>%
71   select(-(duration:listed_in))
72
73 print("--- Dataset with range 'duration' to 'listed_in' dropped ---")
74 print(names(dropped_range))
75
75:1 (Untitled)
R Script

```

OUTPUT:

```

Source
Console Terminal Background Jobs
R - R4.1.2 - ~/
> #7. Selecting and dropping variables using select() in R.
> # =====
> # 1. IMPORT DATASET
> # =====
> library(dplyr) # select() is part of the dplyr package
> # Import the csv file
> housing <- read_csv("R/netflix_titles.csv")
Rows: 8807 Columns: 12
-- Column specification --
Delimiter: "
chr (11): show_id, type, title, director, cast, country, date_added, rating, duration, liste...
dbl (1): release_year

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.
> print("--- Original Dataset (First 3 rows) ---")
[1] "--- Original Dataset (First 3 rows) ---"
> print(head(housing, 3))
# A tibble: 3 x 12
  show_id type title director cast country date_added release_year rating duration listed_in
<chr> <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <chr> <chr> <chr>
1 s1 Movie Dick~ Kirsten~ NA united- September~ 2020 PG-13 90 min Document~
2 s2 TV Show Bloo~ NA Ama~ South~ September~ 2021 TV-MA 2 Seaso~ Internat~
3 s3 TV Show Gang~ Julien~ Sami~ NA September~ 2021 TV-MA 1 Season Crime TV~

# i 1 more variable: description<chr>
> # Method A: Select specific columns by name
> # Scenario: We only want the crime rate (crim), Rooms (rm), and value (medv)
> # (Mapped to Netflix: type, title, release_year)
> selected_cols <- housing %>%
+   select(type, title, release_year)
> print("--- Selected Specific Columns ---")
[1] "--- Selected Specific Columns ---"
> print(head(selected_cols, 3))
# A tibble: 3 x 3
  type title release_year

```

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

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Background Jobs

R - R 4.1.2 - ~/
type title release_year
<chr> <chr> <dbl>
1 Movie Dick Johnson Is Dead 2020
2 TV show Blood & water 2021
3 TV show ganglands 2021
> # Method B: Select a range of adjacent columns
> # Scenario: Select everything from 'crim' to 'nox'
> # (Mapped to Netflix: type:title:rating)
> range_cols <- housing %>%
+ select(type:rating)
> print("--- Selected Range of Columns ---")
[1] "--- Selected Range of Columns ---"
> print(head(range_cols, 3))
# A tibble: 3 x 8
type title director cast country date_added release_year rating
<chr> <chr> <chr> <chr> <chr> <dbl> <chr>
1 Movie Dick Johnson Is Dead Kirsten Johnson NA United- September~ 2020 PG-13
2 TV show Blood & water NA Ama Qamat~ South ~ September~ 2021 TV-MA
3 TV show ganglands Julien Leclercq Sami Boua~ NA September~ 2021 TV-MA
> # Method C: Select using helper functions (e.g., starts_with)
> # Scenario: Select columns that start with "r" (rm, rad)
> # (Netflix columns starting with 'r' include: release_year, rating)
> starts_with_r <- housing %>%
+ select(starts_with("r"))
> print("--- Selected columns starting with 'r' ---")
[1] "--- Selected columns starting with 'r' ---"
> print(head(starts_with_r, 3))
# A tibble: 3 x 2
release_year rating
<dbl> <chr>
1 2020 PG-13
2 2021 TV-MA
3 2021 TV-MA
> # Method A: Drop a single specific column
> # Scenario: Remove the 'chas'
```

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Background Jobs

R - R 4.1.2 - ~/
1 show_id
2 2021 TV-MA
3 2021 TV-MA
> # Method A: Drop a single specific column
> # Scenario: Remove the 'chas'
> # (Mapped to Netflix: drop 'country')
> dropped_one <- housing %>%
+ select(-country)
> print("--- Dataset with 'country' dropped ---")
[1] "--- Dataset with 'country' dropped ---"
> print(names(dropped_one)) # Printing names to verify it's gone
[1] "release_year" "type" "title" "director" "cast" "date_added"
[7] "duration" "listed_in" "description"
> # Method B: Drop multiple columns
> # Scenario: Remove 'zn' and 'indus'
> # (Mapped to Netflix: drop 'cast' and 'director')
> dropped_multiple <- housing %>%
+ select(-cast, -director)
> print("--- Dataset with 'cast' and 'director' dropped ---")
[1] "--- Dataset with 'cast' and 'director' dropped ---"
> print(names(dropped_multiple))
[1] "show_id" "type" "title" "country" "date_added" "release_year"
[7] "rating" "duration" "listed_in" "description"
> # Method C: Drop a range of columns
> # Scenario: Remove everything from 'age' to 'tax'
> # (Mapped to Netflix: duration - listed_in)
> dropped_range <- housing %>%
+ select(-(duration:listed_in))
> print("--- Dataset with range 'duration' to 'listed_in' dropped ---")
[1] "--- Dataset with range 'duration' to 'listed_in' dropped ---"
> print(names(dropped_range))
[1] "show_id" "type" "title" "director" "cast" "country"
[7] "date_added" "release_year" "rating" "description"
>
```