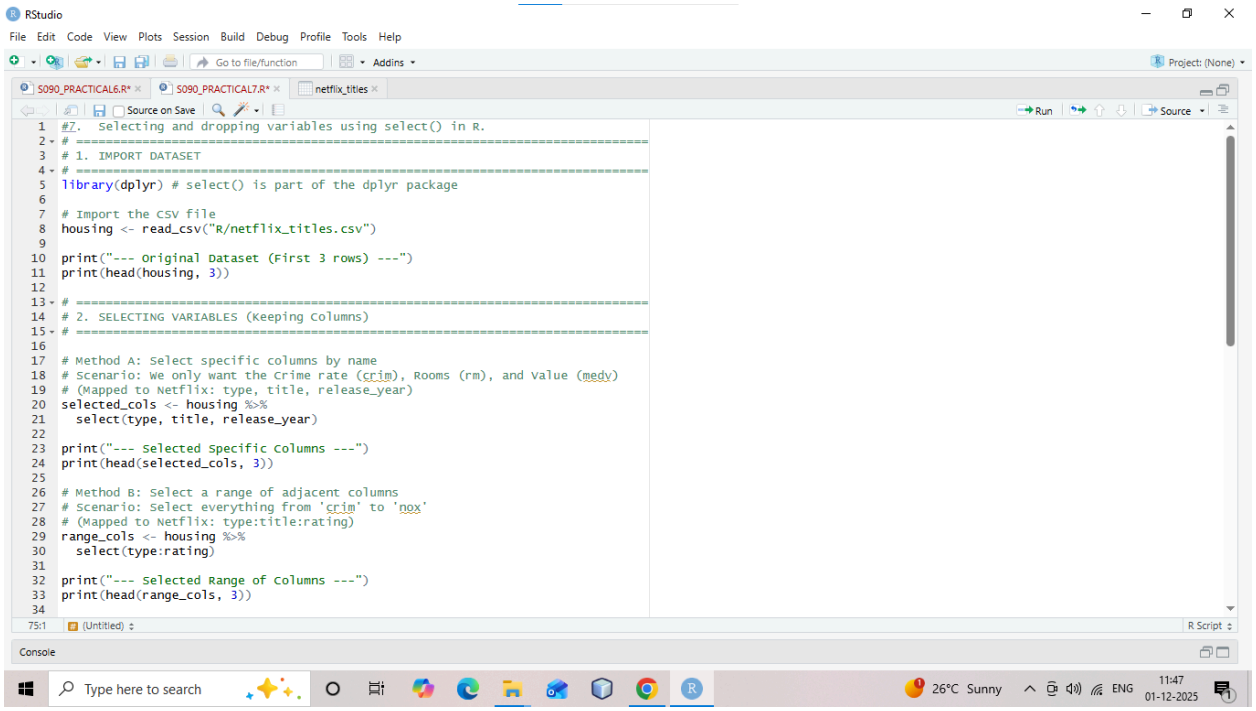
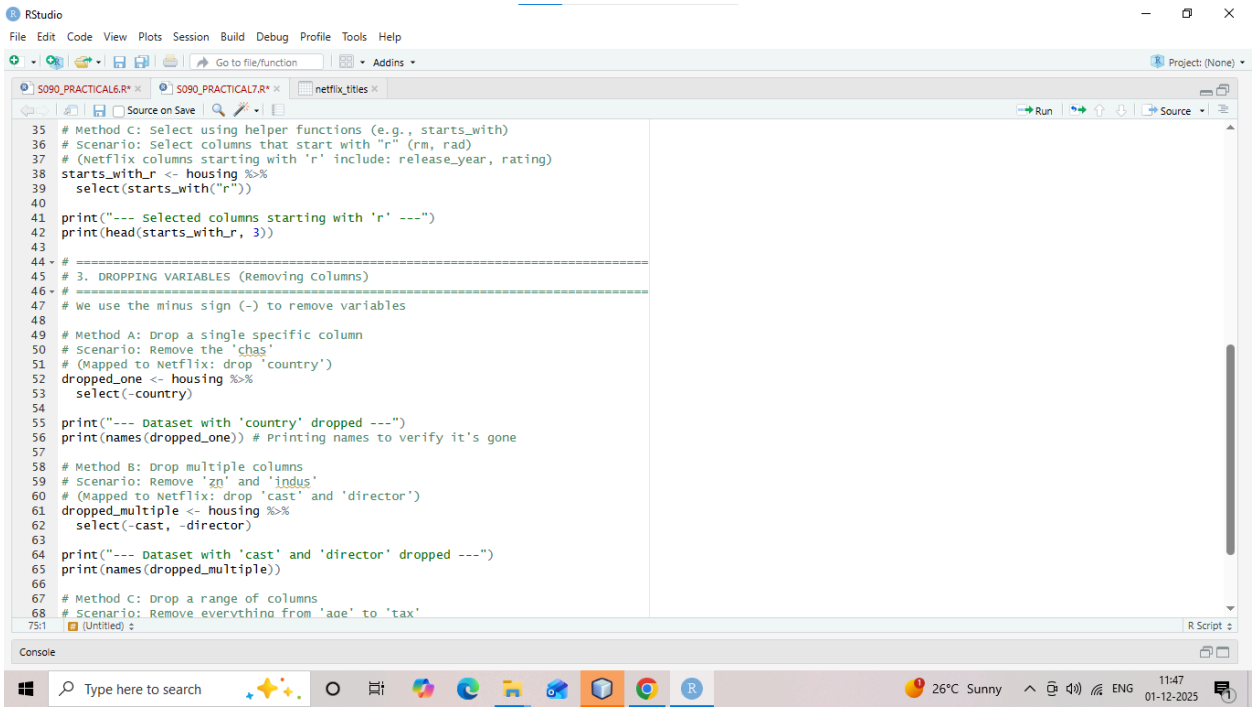
SHETH L.U.J. AND SIR M.V. COLLEGE

PRACTICAL NO 7

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



```r
#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")

print("--- Original Dataset (First 3 rows) ---")
print(head(housing, 3))

# ================================================================================
# 2. SELECTING VARIABLES (Keeping Columns)
# ================================================================================

# 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 ---")
print(head(selected_cols, 3))

# 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 ---")
print(head(range_cols, 3))
```

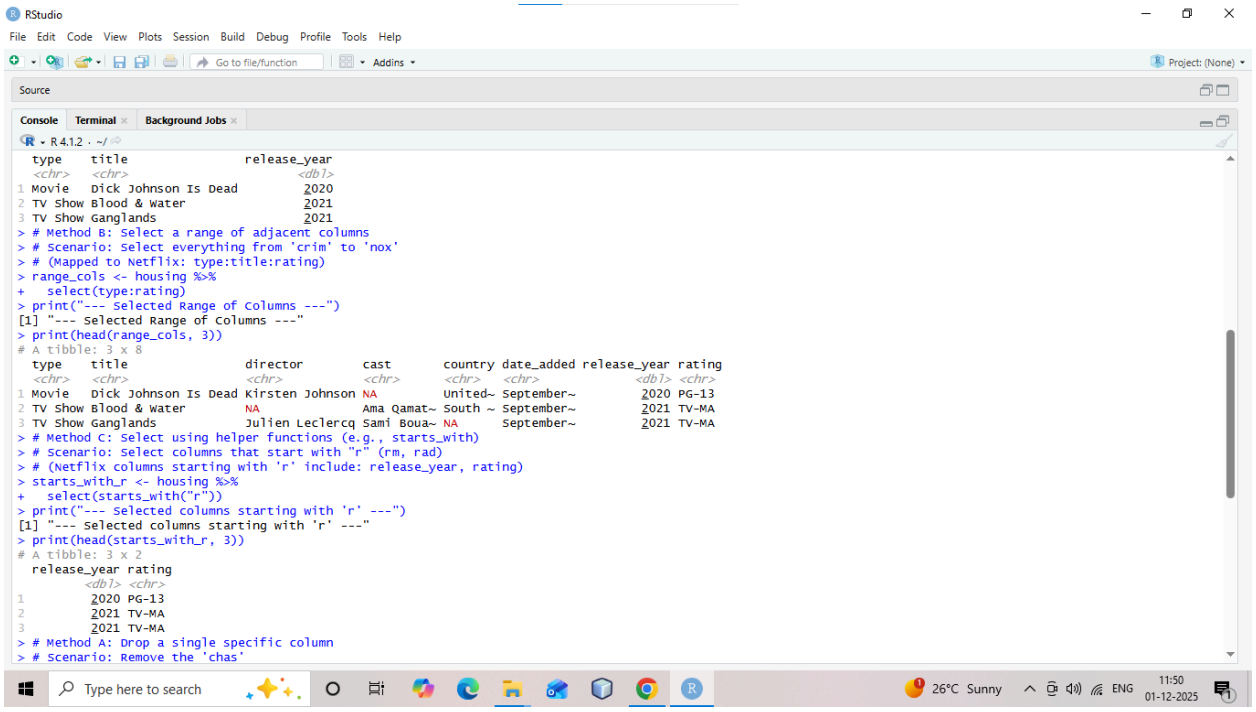


```r
# Method C: Select using helper functions (e.g., starts_with)
# Scenario: Select columns that start with "r" (rm, rad)
# (Netflix columns starting with 'r' include: release_year, rating)
starts_with_r <- housing %>%
  select(starts_with("r"))

print("--- Selected columns starting with 'r' ---")
print(head(starts_with_r, 3))

# ================================================================================
# 3. DROPPING VARIABLES (Removing Columns)
# ================================================================================
# We use the minus sign (-) to remove variables

# 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 ---")
print(names(dropped_one)) # Printing names to verify it's gone

# 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 ---")
print(names(dropped_multiple))

# Method C: Drop a range of columns
# Scenario: Remove everything from 'age' to 'tax'
```

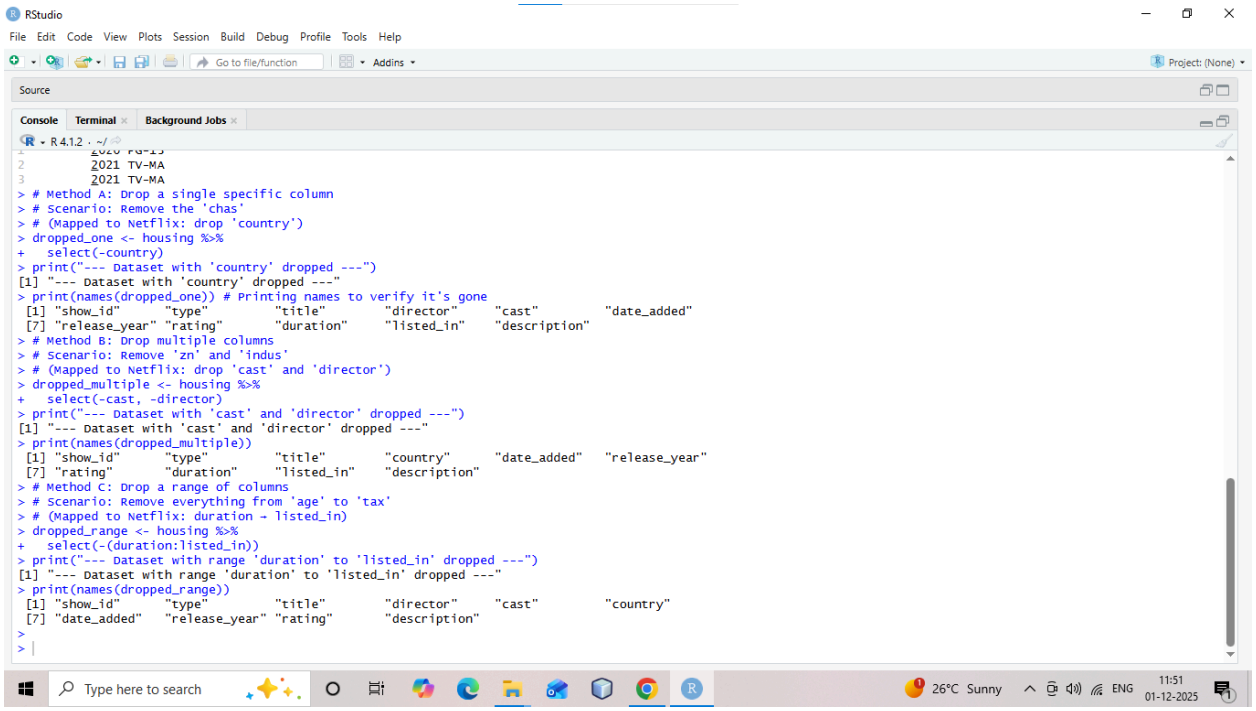
```
print(head(starts_with_r, 3))

# ================================================================================
# 3. DROPPING VARIABLES (Removing Columns)
# ================================================================================
# We use the minus sign (-) to remove variables

# 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 ---")
print(names(dropped_one)) # Printing names to verify it's gone

# 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 ---")
print(names(dropped_multiple))

# 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 ---")
print(names(dropped_range))
```

OUTPUT:

```
> #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
```



```
  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>   <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'
```



```
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] "show_id"      "type"         "title"        "director"     "cast"         "date_added"
 [7] "release_year" "rating"       "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"
>
>
```