Conquering the Realm of Logic: A Quick Guide to Logical Operators in R

Introduction

- Logical operators are the gatekeepers of truth in R, allowing you to combine conditions and build complex decision-making structures.
- Mastering them unlocks the power of conditional statements, loops, and data manipulation tasks.
- This guide will equip you with the knowledge and practice to handle these mighty tools with confidence.

The Binary Universe of TRUE and FALSE

- Before diving into operators, remember R's fundamental truth values: TRUE and FALSE.
- These form the bedrock of any logical expression.
- Statements like 5 > 3 evaluate to TRUE, while 2 + 2 = 5 gives a resounding FALSE

KIRCT

The Big Three: AND, OR, and NOT:

These are the workhorses of R logic:

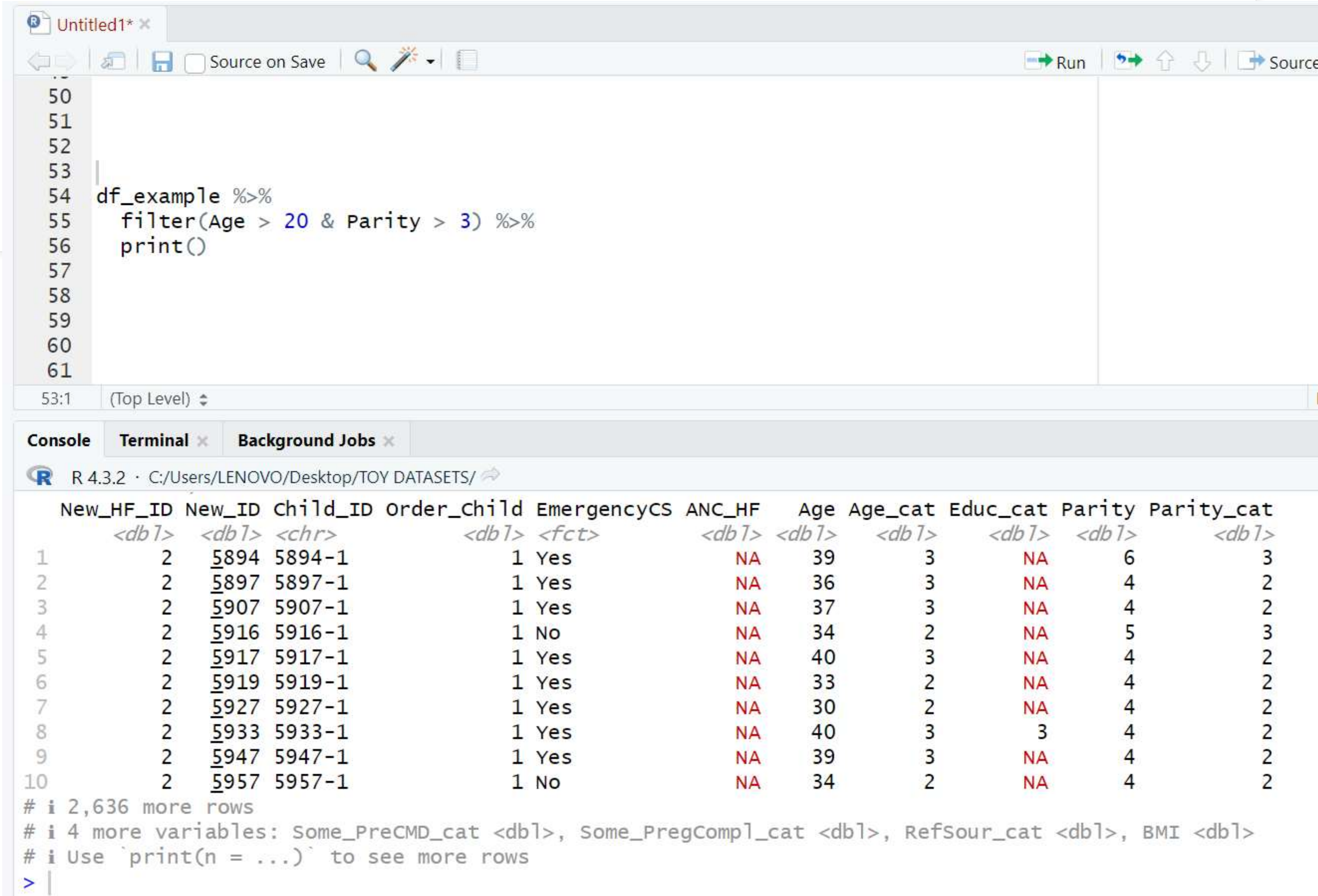
- AND (&): Returns TRUE only if both conditions are TRUE. (TRUE & TRUE = TRUE).
- OR (|): More lenient than AND, OR returns TRUE if at least one condition is TRUE. (TRUE | FALSE = TRUE).
- NOT (!): The rebel of the group, NOT flips the truth value. !TRUE becomes FALSE, and vice versa. It's like a double negative in logic, turning a statement inside out.

Putting them to Work: Examples and Practice

Let's see these operators in action

- Filtering Data: Using the dataset df_example. You can combine logical expressions to find participants who Age both above 20 and Parity > 3:

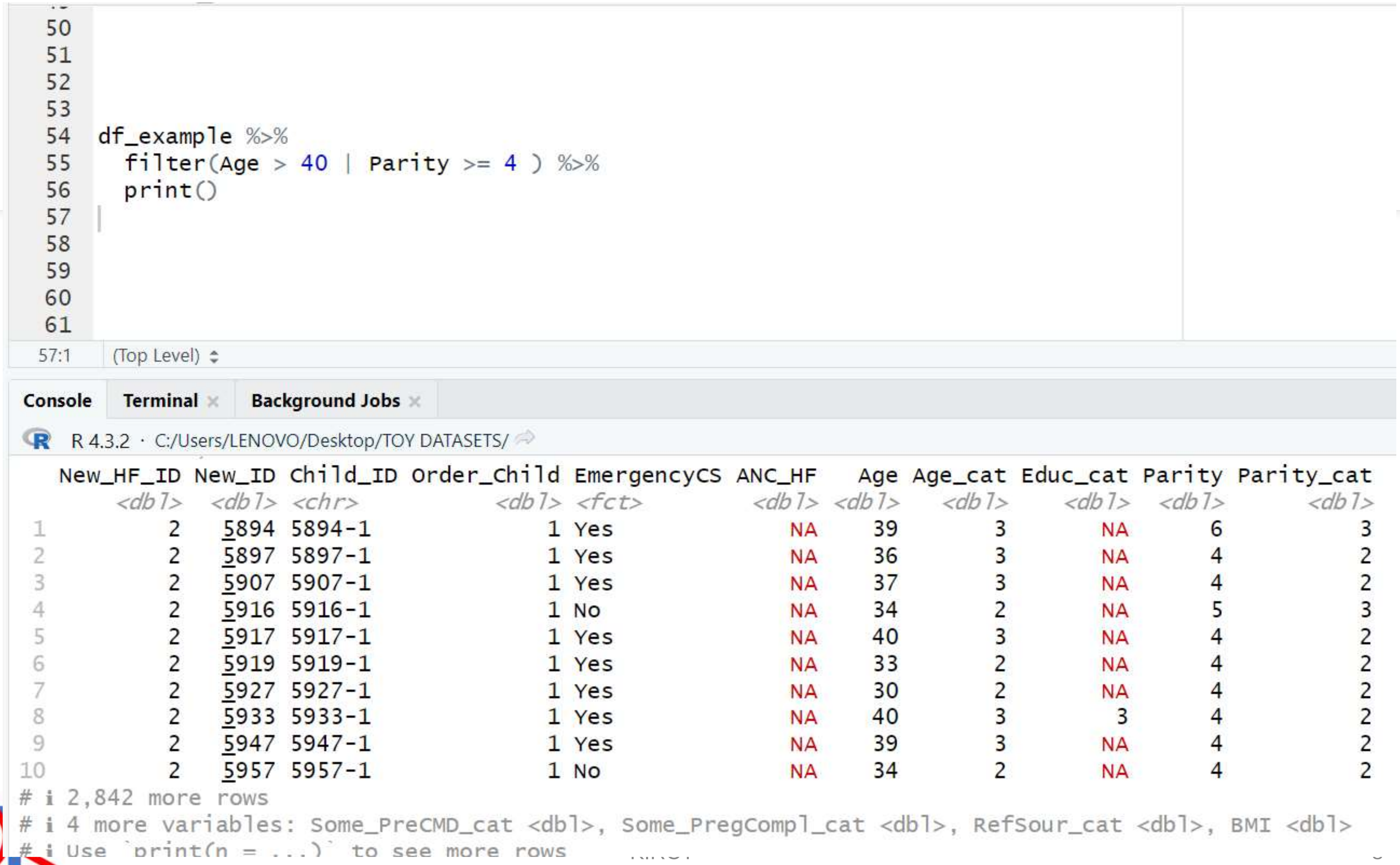
We use pipe operator and filter function to observation the have age >20 and Parity >3



```
df_example %>%
  filter(Age > 20 & Parity > 3) %>%
  print()
```

```
   New_HF_ID New_ID Child_ID Order_Child EmergencyCS ANC_HF   Age Age_cat Educ_cat Parity Parity_cat
       <dbl>  <dbl> <chr>          <dbl> <fct>        <dbl> <dbl>   <dbl>    <dbl>  <dbl>      <dbl>
 1         2   5894 5894-1             1 Yes             NA    39       3       NA      6          3
 2         2   5897 5897-1             1 Yes             NA    36       3       NA      4          2
 3         2   5907 5907-1             1 Yes             NA    37       3       NA      4          2
 4         2   5916 5916-1             1 No              NA    34       2       NA      5          3
 5         2   5917 5917-1             1 Yes             NA    40       3       NA      4          2
 6         2   5919 5919-1             1 Yes             NA    33       2       NA      4          2
 7         2   5927 5927-1             1 Yes             NA    30       2       NA      4          2
 8         2   5933 5933-1             1 Yes             NA    40       3        3      4          2
 9         2   5947 5947-1             1 Yes             NA    39       3       NA      4          2
10         2   5957 5957-1             1 No              NA    34       2       NA      4          2
# ℹ 2,636 more rows
# ℹ 4 more variables: Some_PreCMD_cat <dbl>, Some_PregCompl_cat <dbl>, RefSour_cat <dbl>, BMI <dbl>
# ℹ Use `print(n = ...)` to see more rows
>
```

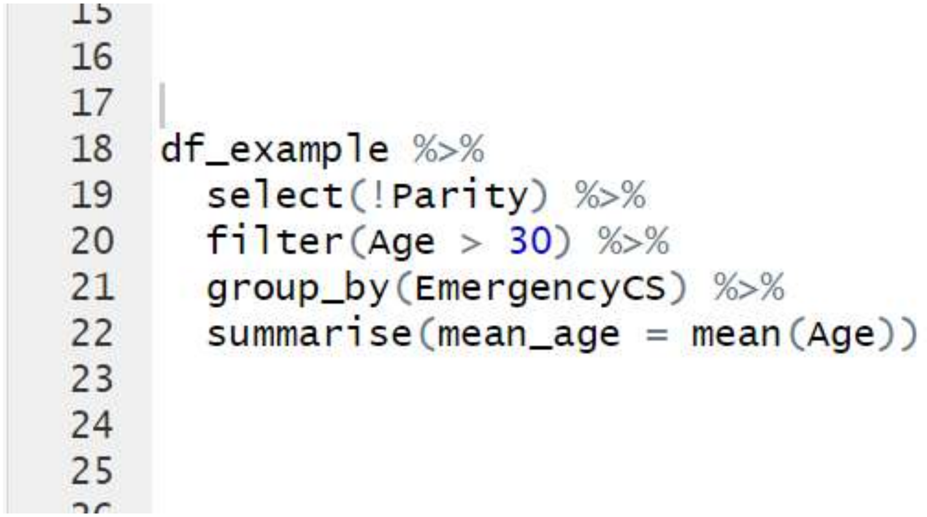
- Conditional Statements: Building an "age verification" script? Use OR to check ID or age:



```r
df_example %>%
  filter(Age > 40 | Parity >= 4 ) %>%
  print()
```

```
   New_HF_ID New_ID Child_ID Order_Child EmergencyCS ANC_HF   Age Age_cat Educ_cat Parity Parity_cat
       <dbl>  <dbl> <chr>          <dbl> <fct>        <dbl> <dbl>   <dbl>    <dbl>  <dbl>      <dbl>
 1         2   5894 5894-1             1 Yes             NA    39       3       NA      6          3
 2         2   5897 5897-1             1 Yes             NA    36       3       NA      4          2
 3         2   5907 5907-1             1 Yes             NA    37       3       NA      4          2
 4         2   5916 5916-1             1 No              NA    34       2       NA      5          3
 5         2   5917 5917-1             1 Yes             NA    40       3       NA      4          2
 6         2   5919 5919-1             1 Yes             NA    33       2       NA      4          2
 7         2   5927 5927-1             1 Yes             NA    30       2       NA      4          2
 8         2   5933 5933-1             1 Yes             NA    40       3        3      4          2
 9         2   5947 5947-1             1 Yes             NA    39       3       NA      4          2
10         2   5957 5957-1             1 No              NA    34       2       NA      4          2
# ℹ 2,842 more rows
# ℹ 4 more variables: Some_PreCMD_cat <dbl>, Some_PregCompl_cat <dbl>, RefSour_cat <dbl>, BMI <dbl>
# ℹ Use `print(n = ...)` to see more rows
```

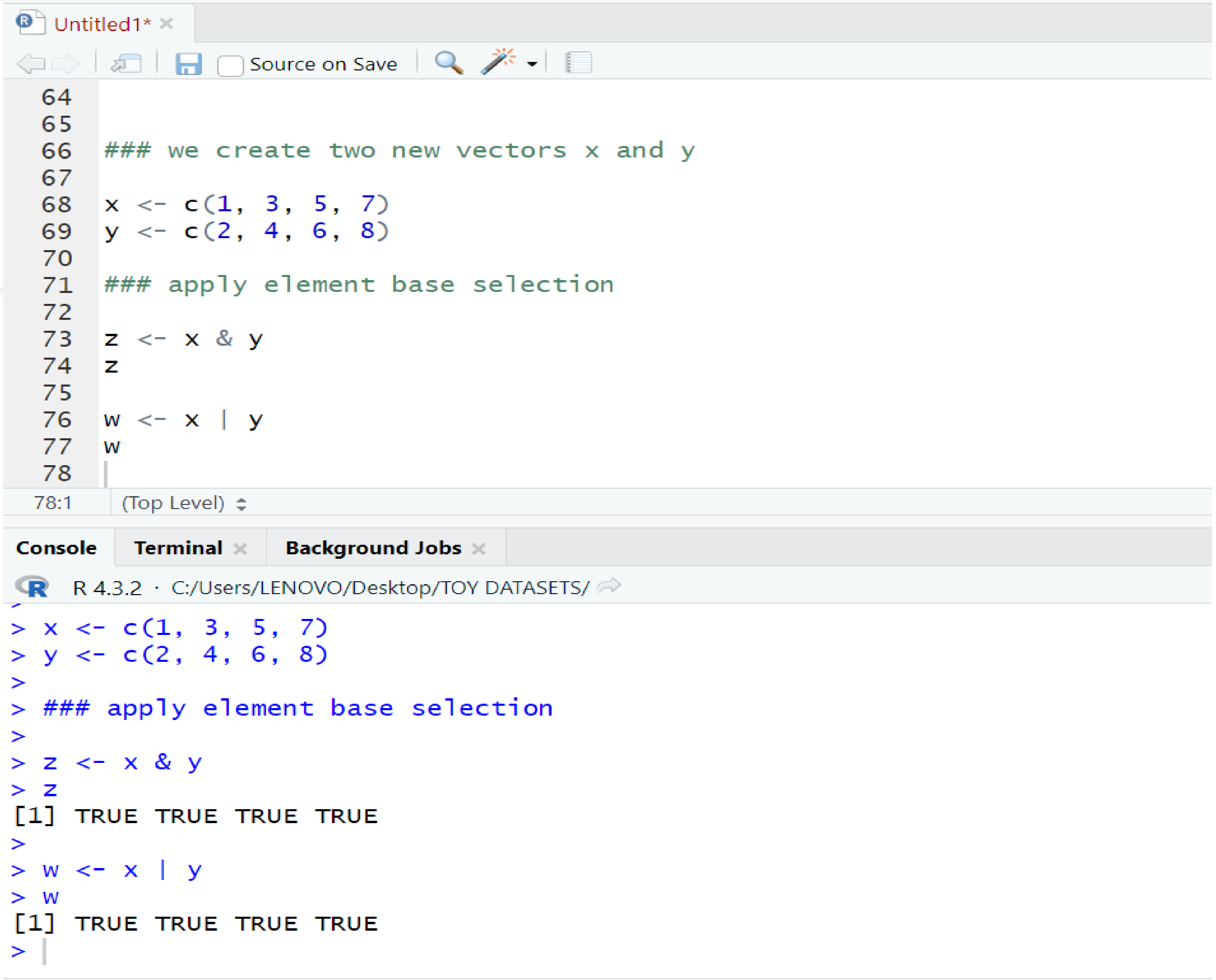
- Negating Results: Want to exclude specific values from a data analysis? NOT comes in handy:



```
df_example %>%
  select(!Parity) %>%
  filter(Age > 30) %>%
  group_by(EmergencyCS) %>%
  summarise(mean_age = mean(Age))
```

Element-wise AND and OR

- R offers handy operators (& and |) for element-wise comparisons within vectors.
- For example:



```r

### we create two new vectors x and y

x <- c(1, 3, 5, 7)
y <- c(2, 4, 6, 8)

### apply element base selection

z <- x & y
z

w <- x | y
w

```

```
> x <- c(1, 3, 5, 7)
> y <- c(2, 4, 6, 8)
>
> ### apply element base selection
>
> z <- x & y
> z
[1] TRUE TRUE TRUE TRUE
>
> w <- x | y
> w
[1] TRUE TRUE TRUE TRUE
>
```

Comparison Operators

- **(<, <=, >, >=, ==, !=):**

- These operators are used to compare values and return logical vectors indicating the result of the comparison.