

## ACADGILD ASSIGNMENT 4.1

```
1. df1 = data.frame(CustId = c(1:6), Product = c(rep("TV", 3),
rep("Radio", 3))) df2 = data.frame(CustId = c(2, 4, 6), State =
c(rep("Texas", 2), rep("NYC", 1))) df1 #left table df2 #right
table
```

For the above given data frames and tables perform the following operations:

- Return only the rows in which the left table have match.
- Returns all rows from both tables, join records from the left which have matching keys in the right table.
- Return all rows from the left table, and any rows with matching keys from the right table.
- Return all rows from the right table, and any rows with matching keys from the left table.

### Answer:

```
df1 <-
data.frame(CustomerId=1:6,Product=c(rep('Toaster',3L),rep('Ra
dio',3L)));
df2 <-
data.frame(CustomerId=c(2L,4L,6L,7L),State=c(rep('Alabama',
2L),'Ohio','Texas'));
df1[names(df2)[-1L]] <- df2[match(df1[,1L],df2[,1L]),-1L];
df1;
```

df1;

	CustomerId	Product	State
1	1	Toaster	<NA>
2	2	Toaster	Alabama
3	3	Toaster	<NA>
4	4	Radio	Alabama

```
5      5 Radio <NA>
6      6 Radio Ohio
```

```
> df2;
```

```
CustomerId State
1      2 Alabama
2      4 Alabama
3      6 Ohio
4      7 Texas
```

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for creating data frames `df1` and `df2`, and merging them. Line 5 has a red error icon.
- Environment:** Lists objects in the global environment: `cs2m` (30 obs. of 6 variables), `df1` (6 obs. of 3 variables), `df2` (4 obs. of 2 variables), `dt1` (6 obs. of 2 variables), `dt2` (3 obs. of 2 variables), `grades` (105 obs. of 22 variables), and `iris.rouned` (150 obs. of 5 variables).
- Files:** Shows the file explorer with a search bar and buttons for 'Install' and 'Update'.
- Console:** Displays the output of the R code, showing the structure of `df1` and `df2`.
- Taskbar:** Shows the Windows taskbar with various application icons and system tray information (ENG US, 12:44 PM, 1/7/2019).

```
1 df1 <- data.frame(CustomerId=1:6,Product=c(rep('Toaster',3L),rep('Radio',3L)));
2 df2 <- data.frame(CustomerId=c(2L,4L,6L,7L),State=c(rep('Alabama',2L),'Ohio','Texas'))
3 df1[names(df2)[-1L]] <- df2[match(df1[,1L],df2[,1L]),-1L];
4 df1;
5
6
7
```

```
> dt1 = as.data.table(df1)
> dt2 = as.data.table(df2)
> setkey(dt1, CustomerId)
> setkey(dt2, CustomerId)
> df1 <- data.frame(CustomerId=1:6,Product=c(rep('Toaster',3L),rep('Radio',3L)));
> df2 <- data.frame(CustomerId=c(2L,4L,6L,7L),State=c(rep('Alabama',2L),'Ohio','Texas'))
> df1[names(df2)[-1L]] <- df2[match(df1[,1L],df2[,1L]),-1L];
> df1;
  CustomerId Product State
1         1 Toaster  <NA>
2         2 Toaster Alabama
3         3 Toaster  <NA>
4         4 Radio  Alabama
5         5 Radio  <NA>
6         6 Radio   Ohio
```

```
merge(df1, df2)
```

```
CustomerId State Product
1      2 Alabama Toaster
2      4 Alabama  Radio
3      6  Ohio  Radio
```

```
> merge(df1, df2, by = "CustomerId")
```

```
CustomerId Product State.x State.y
1      2 Toaster Alabama Alabama
2      4  Radio Alabama Alabama
3      6  Radio  Ohio  Ohio
```

2. Perform the below operations on above given data frames and tables:

- Return a long format of the datasets without matching key.
- Keep only observations in df1 that match in df2.
- Drop all observations in df1 that match in df2.

**Answer:**

```
dt1[dt2, nomatch=0L, on = "CustomerId"]
```

```
> dt1[dt2, nomatch=0L, on = "CustomerId"]
```

```
CustomerId Product State
1:      2 Toaster Alabama
2:      4  Radio Alabama
3:      6  Radio  Ohio
```

```
dt2[dt1, nomatch=0L, on = "CustomerId"]
```

```
> dt2[dt1, nomatch=0L, on = "CustomerId"]
```

```
CustomerId State Product
1:      2 Alabama Toaster
2:      4 Alabama  Radio
```

### 3: 6 Ohio Radio

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for creating data frames, merging them, and attempting an anti-join.
- Environment:** Lists objects in the Global Environment: cs2m (30 obs. of 6 variables), df1 (6 obs. of 3 variables), df2 (4 obs. of 2 variables), dt1 (6 obs. of 2 variables), dt2 (3 obs. of 2 variables), grades (105 obs. of 22 variables), and iris.ornamed (150 obs. of 5 variables).
- Files:** Shows the file explorer with 'Install' and 'Update' buttons.
- Console:** Displays the execution of the code and an error message.

**R Code in Source Editor:**

```
1 df1 <- data.frame(CustomerId=1:6, Product=c(rep('Toaster', 3), rep('Radio', 3)));
2 df2 <- data.frame(CustomerId=c(2L, 4L, 6L, 7L), State=c(rep('Alabama', 2L), 'Ohio', 'Texas'));
3 df1[names(df2)[-1L]] <- df2[match(df1[, 1L], df2[, 1L]), -1L];
4 df1;
5 df2;
6 merge(df1, df2)
7 merge(df1, df2, by = "CustomerId")
8 merge(x = df1, y = df2, by = "CustomerId", all = TRUE)
9 merge(x = df1, y = df2, by = "CustomerId", all.x = TRUE)
10 merge(x = df1, y = df2, by = "CustomerId", all.y = TRUE)
11 dt1[dt2, nomatch=0L, on = "CustomerId"]
12 dt2[dt1, nomatch=0L, on = "CustomerId"]
```

**Console Output:**

```
2      4 Radio Alabama Alabama
3      6 Radio  Ohio  Ohio
4      7 <NA>    <NA> Texas
> anti_join(df1, df2)
Error in anti_join(df1, df2) : could not find function "anti_join"
> dt1[dt2, nomatch=0L, on = "CustomerId"]
  CustomerId Product State
1:         2 Toaster Alabama
2:         4 Radio  Alabama
3:         6 Radio   Ohio
> dt2[dt1, nomatch=0L, on = "CustomerId"]
  CustomerId State Product
1:         2 Alabama Toaster
2:         4 Alabama Radio
3:         6  Ohio Radio
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

Source

```
anti_join(df1, df2)
Error in anti_join(df1, df2) : could not find function "anti_join"
> dt1[dt2, nomatch=0L, on = "CustomerId"]
  CustomerId Product State
1:         2 Toaster Alabama
2:         4 Radio Alabama
3:         6 Radio Ohio
> dt2[dt1, nomatch=0L, on = "CustomerId"]
  CustomerId State Product
1:         2 Alabama Toaster
2:         4 Alabama Radio
3:         6 Ohio Radio
> merge(x = df1, y = df2, by = NULL)
  CustomerId.x Product State.x CustomerId.y State.y
1:           1 Toaster <NA>           2 Alabama
2:           2 Toaster Alabama           2 Alabama
3:           3 Toaster <NA>           2 Alabama
4:           4 Radio Alabama           2 Alabama
5:           5 Radio <NA>           2 Alabama
6:           6 Radio Ohio           2 Alabama
7:           1 Toaster <NA>           4 Alabama
8:           2 Toaster Alabama           4 Alabama
9:           3 Toaster <NA>           4 Alabama
10:          4 Radio Alabama           4 Alabama
11:          5 Radio <NA>           4 Alabama
12:          6 Radio Ohio           4 Alabama
13:           1 Toaster <NA>           6 Ohio
14:           2 Toaster Alabama           6 Ohio
15:           3 Toaster <NA>           6 Ohio
16:           4 Radio Alabama           6 Ohio
17:           5 Radio <NA>           6 Ohio
18:           6 Radio Ohio           6 Ohio
19:           1 Toaster <NA>           7 Texas
20:           2 Toaster Alabama           7 Texas
21:           3 Toaster <NA>           7 Texas
22:           4 Radio Alabama           7 Texas
23:           5 Radio <NA>           7 Texas
24:           6 Radio Ohio           7 Texas
>
```

Environment History Connections

Global Environment

Data

cs2m	30 obs. of 6 variables	
df1	6 obs. of 3 variables	
df2	4 obs. of 2 variables	
dt1	6 obs. of 2 variables	
dt2	3 obs. of 2 variables	
grades	105 obs. of 22 variables	
iris.ornamed	150 obs. of 5 variables	

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
<b>User Library</b>		
<input type="checkbox"/> abind	Combine Multidimensional Arrays	1.4-5
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.0
<input type="checkbox"/> BH	Boost C++ Header Files	1.66.0-1
<input type="checkbox"/> bindr	Parametrized Active Bindings	0.1.1
<input type="checkbox"/> bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2
<input type="checkbox"/> bitops	Bitwise Operations	1.0-6
<input type="checkbox"/> car	Companion to Applied Regression	3.0-2
<input type="checkbox"/> carData	Companion to Applied Regression Data Sets	3.0-2
<input type="checkbox"/> cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
<input type="checkbox"/> cli	Helpers for Developing Command Line Interfaces	1.0.1
<input type="checkbox"/> clipr	Read and Write from the System Clipboard	0.4.1
<input type="checkbox"/> colorspace	Color Space Manipulation	1.3-2
<input type="checkbox"/> crayon	Colored Terminal Output	1.3.4
<input type="checkbox"/> curl	A Modern and Flexible Web Client for R	3.2
<input checked="" type="checkbox"/> data.table	Extension of 'data.frame'	1.11.8
<input type="checkbox"/> digest	Create Compact Hash Digests of R Objects	0.6.18

Type here to search

ENG 1:17 PM 1/7/2019