Joining Datasets – Part 1

Example:

Dataset With Scores

ID	Score
15672	800
16892	"Issue"
56749	650
85413	200

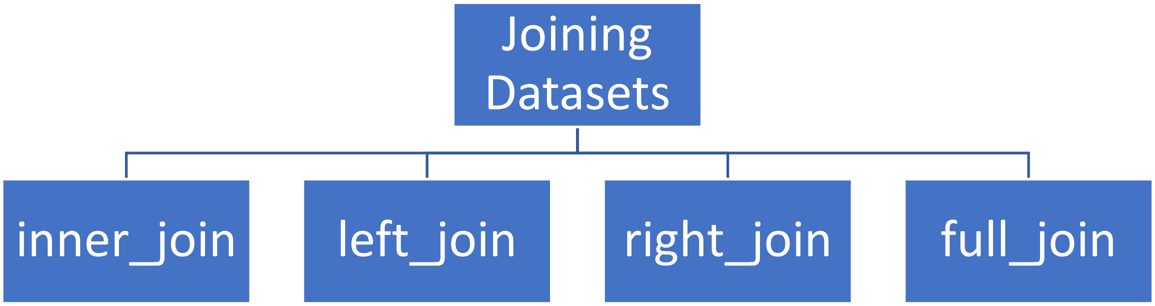
Dataset With Names

ID	Name
15672	Deborah H.
16892	John D.
56749	Errol M.
85413	Juan O.

ID	Name	Score
15672	Deborah H.	800
16892	John D.	"Issue"
56749	Errol M.	650
85413	Juan O.	200

Joins: In the `tidyverse`, joining two datasets together is a way to combine data from different sources based on common variables.

The `dplyr` package within the tidyverse provides functions to perform these joins.



Joining Datasets

inner_join

left_join

right_join

full_join

Inner Join: An inner join combines rows from two datasets where there's a match between the specified variables.

- Rows with no matching values are excluded.
- Inner joins return results if the keys are matched in BOTH tables.

Command Illustration

```
new_dataframe_name <- dataframe_name_1 %>%
     inner_join(dataframe_name_2, c("colname_1" = "colname_2"))
```

For the illustration examples, assume the two following dataframes:

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Illustration_Data_2

First_Name	Last_Name	Gender
Val	Chmerkovskiy	Male
Derek	Hough	Male
Whitney	Carson	Female
Sasha	Farber	Male
Daniella	Karagach	Female
Lindsay	Arnold	Female
Mark	Ballas	Male

Example 1: Perform an inner join between Illustration_Data_1 and Illustration_Data_2

```
example_1 <- Illustration_Data_1 %>%
  inner_join(Illustration_Data_2, c("Name" = "First_Name"))
```


For the illustration examples, assume the two following dataframes:

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Illustration_Data_3

Name	Last_Name	Car
Val	Chmerkovskiy	Mercedes
Val	Chmerkovskiy	Tesla
Val	Chmerkovskiy	Audi
Derek	Hough	Ferrari
Lindsay	Arnold	Tesla
Mark	Ballas	BMW

Example 2: Perform an inner join between Illustration_Data_1 and Illustration_Data_3

```
example_2 <- Illustration_Data_1 %>%
  inner_join(Illustration_Data_3, c("Name" = "Name"))
```


It is always a good idea to carefully check that the number of rows returned by a join operation is what you expected. In particular, you should carefully check for rows in one table that matched to more than one row in the other table.

- Inspect the column by which you are joining.

`nrow(Illustration_Data_1)` **Output:**

`n_distinct(Illustration_Data_1$Name)` **Output:**

`nrow(Illustration_Data_2)` **Output:**

`n_distinct(Illustration_Data_2$First_Name)` **Output:**

- Check how many data values from one dataset are in the other dataset.

```
table(Illustration_Data_1$Name %in% Illustration_Data_2$First_Name)
```

Output:

```
table(Illustration_Data_2$First_Name %in% Illustration_Data_1$Name)
```

Output:

Joining Datasets – Part 2

Left Join

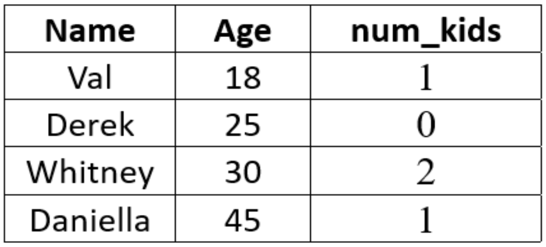
Left Join: includes all rows from the left dataset and the matching rows from the right dataset. If there's no match, the columns from the right dataset will be filled with NA. Here the rows of the first table are always returned, regardless of whether there is a match in the second table.

Command Illustration

```
new_dataframe_name <- dataframe_left %>%
     left_join(dataframe_right, c("colname_1" = "colname_2"))
```

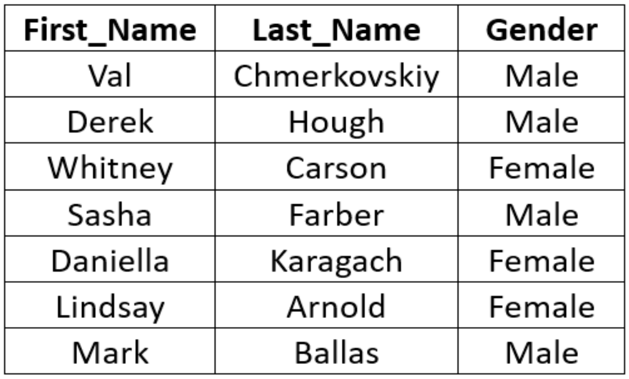
For Example 1, assume the two following dataframes:

Illustration_Data_1



Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Illustration_Data_2



First_Name	Last_Name	Gender
Val	Chmerkovskiy	Male
Derek	Hough	Male
Whitney	Carson	Female
Sasha	Farber	Male
Daniella	Karagach	Female
Lindsay	Arnold	Female
Mark	Ballas	Male

Example 1: Do a left join where Illustration_Data_1 is the left data and Illustration_Data_2 is the right data.

```
example_1 <- Illustration_Data_1 %>%
  left_join(Illustration_Data_2, by = c("Name" = "First_Name"))
```


For Example 2, assume the two following dataframes:

Illustration_Data_2

First_Name	Last_Name	Gender
Val	Chmerkovskiy	Male
Derek	Hough	Male
Whitney	Carson	Female
Sasha	Farber	Male
Daniella	Karagach	Female
Lindsay	Arnold	Female
Mark	Ballas	Male

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Example 2: Do a left join where Illustration_Data_2 is the left data and Illustration_Data_1 is the right data.

```
example_2 <- Illustration_Data_2 %>%
  left_join(Illustration_Data_1, by = c("First_Name" = "Name"))
```


For Example 3, assume the two following dataframes:

Illustration_Data_3

Name	Last_Name	Car
Val	Chmerkovskiy	Mercedes
Val	Chmerkovskiy	Tesla
Val	Chmerkovskiy	Audi
Derek	Hough	Ferrari
Lindsay	Arnold	Tesla
Mark	Ballas	BMW

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Example 3: Do a left join where Illustration_Data_3 is the left data and Illustration_Data_1 is the right data.

```r
example_3 <- Illustration_Data_3 %>%
  left_join(Illustration_Data_1, by = c("Name" = "Name"))
```


For Example 4, assume the two following dataframes:

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Illustration_Data_3

Name	Last_Name	Car
Val	Chmerkovskiy	Mercedes
Val	Chmerkovskiy	Tesla
Val	Chmerkovskiy	Audi
Derek	Hough	Ferrari
Lindsay	Arnold	Tesla
Mark	Ballas	BMW

Example 4: Do a left join where Illustration_Data_1 is the left and Illustration_Data_3 is the right.

```
example_4 <- Illustration_Data_1 %>%
  left_join(Illustration_Data_3, by = c("Name" = "Name"))
```

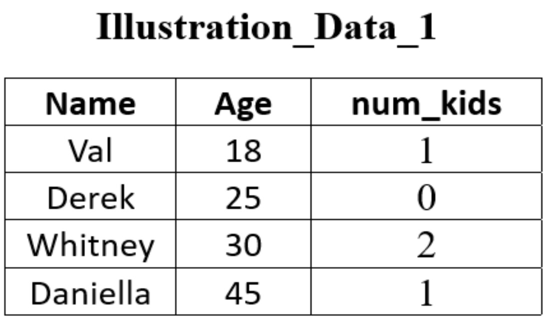

Right Join

Right Join: is the opposite of a left join. It includes all rows from the right dataset and the matching rows from the left dataset. A right join but this is much less common.

Command Illustration

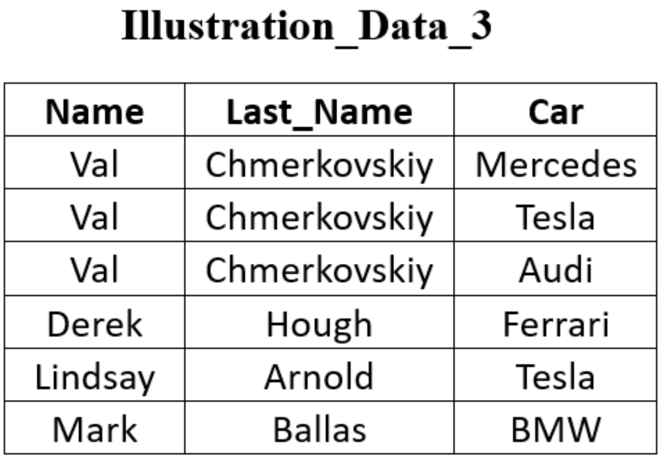
```
new_dataframe_name <- dataframe_left %>%
     right_join(dataframe_right, c("colname_1" = "colname_2"))
```

For Example 5, assume the two following dataframes:



Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1



Illustration_Data_3

Name	Last_Name	Car
Val	Chmerkovskiy	Mercedes
Val	Chmerkovskiy	Tesla
Val	Chmerkovskiy	Audi
Derek	Hough	Ferrari
Lindsay	Arnold	Tesla
Mark	Ballas	BMW

Example 5: Do a right join where Illustration_Data_1 is the left and Illustration_Data_3 is the right.

```r
example_5 <- Illustration_Data_1 %>%
  right_join(Illustration_Data_3, by = c("Name" = "Name"))
```


Summarizing NA's

Example 2 Output:

First_Name	Last_Name	Gender	Age	num_kids
Val	Chmerkovskiy	Male	18	1
Derek	Hough	Male	25	0
Whitney	Carson	Female	30	2
Sasha	Farber	Male	*NA*	*NA*
Daniella	Karagach	Female	45	1
Lindsay	Arnold	Female	*NA*	*NA*
Mark	Ballas	Male	*NA*	*NA*

Example 6: Summarize the NA's for the left joined data from Example 2

```
example_6 <- example_2 %>%
  summarize(num_people = n(),
            num_na = sum(is.na(Age)),
            num_not_na = sum(!is.na(Age)))
```


Joining Datasets – Part 3

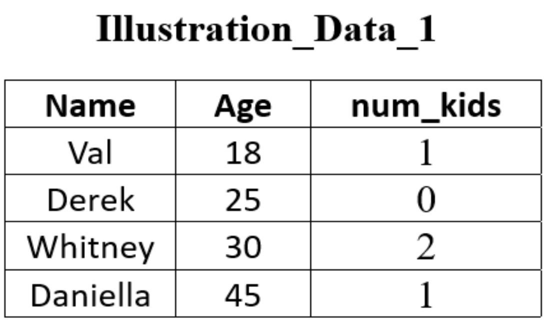
Full Join

Full Join: includes all rows from both datasets. Columns from the dataset with missing values will be filled with NA where there's no match.

Command Illustration

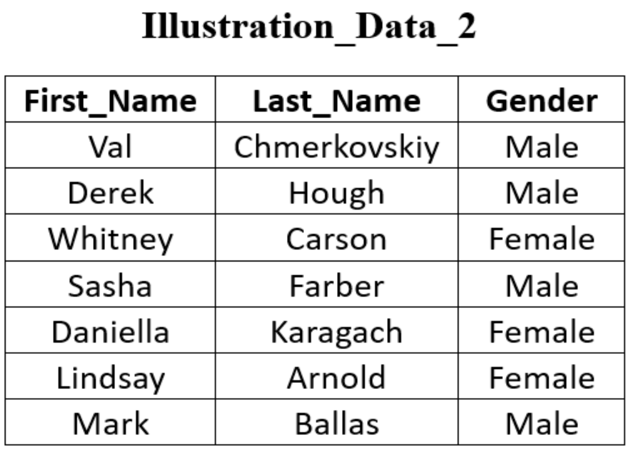
```
new_dataframe_name <- dataframe_1 %>%
     full_join(dataframe_2, c("colname_1" = "colname_2"))
```

For Example 1, assume the two following dataframes:



Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1



Illustration_Data_2

First_Name	Last_Name	Gender
Val	Chmerkovskiy	Male
Derek	Hough	Male
Whitney	Carson	Female
Sasha	Farber	Male
Daniella	Karagach	Female
Lindsay	Arnold	Female
Mark	Ballas	Male

Example 1: Do a full join between Illustration_Data_1 & Illustration_Data_2.

```
example_1 <- Illustration_Data_1 %>%
  full_join(Illustration_Data_2, by = c("Name" = "First_Name"))
```


For Example 2, assume the two following dataframes:

Illustration_Data_1

Name	Age	num_kids
Val	18	1
Derek	25	0
Whitney	30	2
Daniella	45	1

Illustration_Data_3

Name	Last_Name	Car
Val	Chmerkovskiy	Mercedes
Val	Chmerkovskiy	Tesla
Val	Chmerkovskiy	Audi
Derek	Hough	Ferrari
Lindsay	Arnold	Tesla
Mark	Ballas	BMW

Example 2: Do a full join between Illustration_Data_1 & Illustration_Data_3.

```
example_2 <- Illustration_Data_1 %>%
  full_join(Illustration_Data_3, by = c("First_Name" = "Name"))
```


Join by Multiply Columns

Note: You will not be tested on this material

Join by multiple columns: The `by` argument specifies the column(s) that should be used for matching. These join functions work well when the datasets have a shared column containing the same type of data (e.g., IDs or keys). You can use multiple column names to define the matching conditions.

Command Illustration

```
new_dataframe <- name_dataframe_x %>%
         inner_join(name_dataframe_y, by=c("x1" = "y1" ,  "x2" =
"y2"))
```

Example 3: We will manually create two datafames in R. Merge the following two dataframes (emp_df & dept_df) by "dept_id" & "dept_branch_id".

```
example_3 <- emp_df %>%
     inner_join(dept_df,   by   =   c("dept_id"   =   "dept_id",
"dept_branch_id" = "dept_branch_id"))
```

emp_df

emp_id	name	superior_emp_id	dept_id	dept_branch_id
1	Smith	-1	10	101
2	Rose	1	20	102
3	Williams	1	10	101
4	Jones	2	10	101
5	Brown	2	40	104
6	Brown	2	50	105

dept_df

dept_id	dept_branch_id	dept_name
10	101	Finance
20	102	Marketing
30	103	Sales
40	104	IT

When you merge these two dataframes, your output will look like:

emp_id	**name**	**superior_emp_id**	**dept_id**	**dept_branch_id**	**dept_name**
1	Smith	-1	10	101	Finance
2	Rose	1	20	102	Marketing
3	Williams	1	10	101	Finance
4	Jones	2	10	101	Finance
5	Brown	2	40	104	IT