

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

```

{r}
res1 <-
df %>%
  group_by(v1, v2)
res1

```

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

A group is a set of rows with the same values of v1 and v2

```

{r}
res2 <-
res1 %>%
  summarize(
    mean_v3 = mean(v3, na.rm = TRUE),
    max_v3 = max(v3, na.rm = TRUE)
  )
res2

```

V1	V2	mean_v3	max_v3
A	C	1.67	2
A	D	1.5	5
B	E	2.67	3
B	F	3	4

```

{r}
res3 <-
res2 %>%
  ungroup()
res3

```

V1	V2	mean_v3	max_v3
A	C	1.67	2
A	D	1.5	5
B	E	2.67	3
B	F	3	4

We used *summarize()*

Result has one row per group (per unique combination of v1 and v2)

Only the variables defined within the *summarize* operation remain.

Groups are now defined only by v1 (because the combination of unique values of v1 and v2 defines only 1-row sets, not groups).

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

```
...{r}  
res1 <-  
  df %>%  
    group_by(v1, v2)  
res1  
...
```

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

A group is a set of rows with the same values of v1 and v2

```
...{r}  
res2 <-  
  res1 %>%  
    mutate(  
      v3_mean = mean(v3, na.rm = TRUE) %>% round(2),  
      v3_max = max(v3, na.rm = TRUE) %>% round(2),  
      row_nuber = row_number()  
    )  
...
```

V1	V2	V3	mean_v3	max_v3	row_number()
A	C	1	1.67	2	1
A	C	2	1.67	2	2
A	C	2	1.67	2	3
A	D	5	1.5	5	1
A	D	1	1.5	5	2
B	E	3	2.67	3	1
B	E	3	2.67	3	2
B	E	2	2.67	3	3
B	F	3	3	4	1
B	F	2	3	4	2
B	F	4	3	4	3
B	F	3	3	4	4

```
...{r}  
res3 <-  
  res2 %>%  
    ungroup()  
res3  
...
```

V1	V2	V3	mean_v3	max_v3	row_number()
A	C	1	1.67	2	1
A	C	2	1.67	2	2
A	C	2	1.67	2	3
A	D	5	1.5	5	1
A	D	1	1.5	5	2
B	E	3	2.67	3	1
B	E	3	2.67	3	2
B	E	2	2.67	3	3
B	F	3	3	4	1
B	F	2	3	4	2
B	F	4	3	4	3
B	F	3	3	4	4

We used *mutate()*

The number of rows is preserved and operations are applied *group-wise*

(in SQL – PARTITION BY)

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

```
{r}
res1 <-
df %>%
  group_by(v1, v2)
res1
```

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

A group is a set of rows with the same values of v1 and v2

```
{r}
res2 <-
res1 %>%
  filter(
    v3 == max(v3)
  )
res2
```

```
{r}
res3 <-
res2 %>%
  ungroup()
res3
```

V1	V2	V3
A	C	2
A	C	2
A	D	5
B	E	3
B	E	3
B	F	4

V1	V2	V3
A	C	2
A	C	2
A	D	5
B	E	3
B	E	3
B	F	4

We used  
*filter()*

Within each *group*, only the rows where the condition(s) is(are) fulfilled are preserved

Of course, all these operations might be grouped in a single code chunk

Example:

V1	V2	V3
A	C	1
A	C	2
A	C	2
A	D	5
A	D	1
B	E	3
B	E	3
B	E	2
B	F	3
B	F	2
B	F	4
B	F	3

```
```{r}
res <-
  df %>%
    group_by(v1, v2) %>%
    mutate(
      v3_mean = mean(v3, na.rm = TRUE) %>% round(2),
      v3_max = max(v3, na.rm = TRUE) %>% round(2),
      row_number = row_number()
    ) %>%
    ungroup()
res
```
```

| V1 | V2 | V3 | mean_v3 | max_v3 | row_number() |
|----|----|----|---------|--------|--------------|
| A  | C  | 1  | 1.67    | 2      | 1            |
| A  | C  | 2  | 1.67    | 2      | 2            |
| A  | C  | 2  | 1.67    | 2      | 3            |
| A  | D  | 5  | 1.5     | 5      | 1            |
| A  | D  | 1  | 1.5     | 5      | 2            |
| B  | E  | 3  | 2.67    | 3      | 1            |
| B  | E  | 3  | 2.67    | 3      | 2            |
| B  | E  | 2  | 2.67    | 3      | 3            |
| B  | F  | 3  | 3       | 4      | 1            |
| B  | F  | 2  | 3       | 4      | 2            |
| B  | F  | 4  | 3       | 4      | 3            |
| B  | F  | 3  | 3       | 4      | 4            |

It is paramount that you understand every single step of the code you are executing! Particularly as a beginner!