Window Functions in R: : **CHEAT SHEET**

Basics

A **window function** is a variation on an aggregation function, where it takes n inputs and instead of returning a single value returns n values. The output of a window function depends on all its input values

Ranking & Ordering Functions

- row_number ()

Creates an identification number or label for each observation by a grouping variable.

```
dt <- student_data %>% arrange(DOE) %>%
mutate(ID = row_number())
```

Student	DOE	ID
<chr>	<date>	<int>
Rita	2022-07-08	1
Brian	2022-07-11	2
Alex	2022-07-12	3

- min_rank ()

Used for ranking each observation by a grouping variable

```
minrank <- df %>% group_by(Subject) %>%
mutate(minrnk = min_rank(desc(Marks)))
```

Student	Subject	minrnk
Rita	Science	3
Brian	Science	2
Alex	Science	1
Rita	Maths	1
Brian	Maths	1
Alex	Maths	3

- dense_rank ()

Similar to min_rank() except that there are no gaps between ranks

```
denrank <- df %>% group_by(Subject) %>%
mutate(denrnk = dense_rank(desc(Marks)))
```

- percent_rank ()

Returns a relative rank/ percentile of rows within a window partition

```
perrank <- df %>% group_by(Subject) %>%
mutate(perrnk = percent_rank(desc(Marks)))
```

- ntile()

Returns a coarse rank by dividing the data into n evenly sized buckets

```
nt <- df %>% group_by(Subject) %>%
mutate(nt = ntile(desc(Marks),2))
```

Student	Subject	Marks	nt
Rita	Science	78	2
Brian	Science	86	1
Alex	Science	90	1
Rita	Maths	92	1
Brian	Maths	92	1
Alex	Maths	79	2

- cume_dist()

Returns the cumulative distribution of values within a window partition. It is computed by: cume_dist(x) = number of values uptil x / N;

```
cd <- df %>% group_by(Subject) %>%
mutate(cd = cume_dist(desc(Marks)))
```

Student	Subject	Marks	cd
Rita	Science	78	1.0000000
Brian	Science	86	0.6666667
Alex	Science	90	0.3333333
Rita	Maths	92	0.6666667
Brian	Maths	92	0.6666667
Alex	Maths	79	1.0000000

Offset Functions

- lead()

Introduces an offset such that the returned value is the next value of the input variable

```
lead <- df %>% mutate(ld <- lead(Marks,1,NA))
```

Student	Subject	Marks	ld
Rita	Science	78	86
Brian	Science	86	90
Alex	Science	90	92
Rita	Maths	92	92
Brian	Maths	92	79
Alex	Maths	79	NA

- lag()

introduces an offset such that the returned value is the previous value of the input variable

```
lag <- df %>% mutate(lg <- lag(Marks,1,NA))
```

Student	Subject	Marks	lg
Rita	Science	78	NA
Brian	Science	86	78
Alex	Science	90	86
Rita	Maths	92	90
Brian	Maths	92	92
Alex	Maths	79	92

Cumulative Aggregate Functions

- cumsum()

Returns the cumulative sum of the elements of the input vector or column of a dataframe within the window partition

```
cumsum <- df %>% group_by(Subject) %>%
mutate(cumsum = cumsum(Marks))
```

Student	Subject	Marks	cumsum
Rita	Science	78	78
Brian	Science	86	164
Alex	Science	90	254
Rita	Maths	92	92
Brian	Maths	92	184
Alex	Maths	79	263

- cummin()

Returns the cumulative sum of the elements of the input vector or column of a dataframe within the window partition

```
cummin <- df %>% group_by(Subject) %>%
mutate(cummin = cummin(Marks))
```

- cummean()

returns the cumulative mean of the elements of the input vector or column of a dataframe within the window partition

```
cummean <- df %>% group_by(Subject) %>%
mutate(cummean = cummean(Marks))
```

- cumall()

Checks whether the first data element satisfies the logical condition. If yes, then it returns TRUE. Then it checks whether the first AND second element satisfies the logical condition. This occurs cumulatively till the last data element

```
cumall <- df %>% group_by(Subject) %>%
mutate(cumall = cumall(Marks < 90))
```

Student	Subject	Marks	cumall
Rita	Science	78	TRUE
Brian	Science	86	TRUE
Alex	Science	90	FALSE
Rita	Maths	92	FALSE
Brian	Maths	92	FALSE
Alex	Maths	79	FALSE

- cumany()

checks whether the first data element satisfies the logical condition. If yes, then it returns TRUE. Then it checks whether the first OR second element satisfies the logical condition. This occurs cumulatively till the last data element

```
cumany <- df %>% group_by(Subject) %>%
mutate(cumany = cumany(Marks == 90))
```

Student	Subject	Marks	cumany
Rita	Science	78	FALSE
Brian	Science	86	FALSE
Alex	Science	90	TRUE
Rita	Maths	92	FALSE
Brian	Maths	92	FALSE
Alex	Maths	79	FALSE

Student Table

Student	Subject	Marks	DOE
Rita	Science	78	2022-07-08
Brian	Science	86	2022-07-11
Alex	Science	90	2022-07-12
Rita	Maths	92	2022-07-08
Brian	Maths	92	2022-07-11
Alex	Maths	79	2022-07-12