

# Data Wrangling – Part 3

# mutate() Function

## Command Illustration

```
new_dataframe_name <- dataframe_name %>%  
  mutate(new_column_name = computation you  
        want to do)
```

# mutate() Function

## Illustration\_Data

Name	Age	total_Income	var_1	var_2	var_3	zipcode	honesty	cat_total
Val	18	18000	apple	carrots	elephant	60001	agree	0
Derek	25	25000	grapes	carrots	tiger	60073	disagree	1
Whitney	30	30000	bananas	carrots	lion	60109	disagree	2
Sasha	40	40000	peaches	carrots	rabbit	60111	disagree	1
Daniella	45	45000	bananas	carrots	shark	60155	agree	1

**Example 1:** Add a new variable "income\_per\_month" calculated from "total\_Income"

```
example_1 <- Illustration_Data %>%  
  mutate(income_per_month = total_Income / 12)
```

**Output:** The output is a dataframe that looks like this

Name	Age	total_Income	var_1	var_2	var_3	zipcode	honesty	cat_total	income_per_month
Val	18	18000	apple	carrots	elephant	60001	agree	0	1500
Derek	25	25000	grapes	carrots	tiger	60073	disagree	1	2083
Whitney	30	30000	bananas	carrots	lion	60109	disagree	2	2500
Sasha	40	40000	peaches	carrots	rabbit	60111	disagree	1	3333
Daniella	45	45000	bananas	carrots	shark	60155	agree	1	3750

**Example 2:** Create a new variable called "status" based on "age"

```
example_2 <- Illustration_Data %>%  
  mutate(status = ifelse(Age > 30, "Older", "Younger"))
```

**Output:** The output is a dataframe that looks like this

Name	Age	total_Income	var_1	var_2	var_3	zipcode	honesty	cat_total	Status
Val	18	18000	apple	carrots	elephant	60001	agree	0	Younger
Derek	25	25000	grapes	carrots	tiger	60073	disagree	1	Younger
Whitney	30	30000	bananas	carrots	lion	60109	disagree	2	Younger
Sasha	40	40000	peaches	carrots	rabbit	60111	disagree	1	Older
Daniella	45	45000	bananas	carrots	shark	60155	agree	1	Older

```
table(example_2$Age, example_2$status)
```

```
> table(example_2$Age, example_2$status)
```

	older	Younger
18	0	1
25	0	1
30	0	1
40	1	0
45	1	0
.		

```
case_when(  
  boolean expression ~ value_1,  
  boolean expression ~ value_2,  
  ...,  
  TRUE ~ default_value  
)
```

**Example 3:** Create a new variable "group" based on "age". If the person's age is smaller than 30, they are "Young", if their age is between 30 and 40 (inclusive), then they are "Middle-Aged", if their age is greater than 40, then they are "old".

```
example_3 <- Illustration_Data %>%  
  mutate(group = case_when(  
    Age < 30 ~ "Young",  
    Age >= 30 & Age <= 40 ~ "Middle-Aged",  
    Age > 40 ~ "Old"  
)
```

Name	Age	total_Income	var_1	var_2	var_3	zipcode	honesty	cat_total	group
Val	18	18000	apple	carrots	elephant	60001	agree	0	Young
Derek	25	25000	grapes	carrots	tiger	60073	disagree	1	Young
Whitney	30	30000	bananas	carrots	lion	60109	disagree	2	Middle-Aged
Sasha	40	40000	peaches	carrots	rabbit	60111	disagree	1	Middle-Aged
Daniella	45	45000	bananas	carrots	shark	60155	agree	1	Old

**Example 4:** Create a new variable "group" based on "age". If the person's age is smaller than 30, they are "Young", if their age is between 30 and 40 (inclusive), then they are "Middle-Aged", anything else, simply have it be "No Category".

```
example_4 <- Illustration_Data %>%  
  mutate(group = case_when(  
    Age < 30 ~ "Young",  
    Age >= 30 & Age <= 40 ~ "Middle-aged",  
    TRUE ~ "No Category"  
)
```

Name	Age	total_Income	var_1	var_2	var_3	zipcode	honesty	cat_total	group
Val	18	18000	apple	carrots	elephant	60001	agree	0	Young
Derek	25	25000	grapes	carrots	tiger	60073	disagree	1	Young
Whitney	30	30000	bananas	carrots	lion	60109	disagree	2	Middle-Aged
Sasha	40	40000	peaches	carrots	rabbit	60111	disagree	1	Middle-Aged
Daniella	45	45000	bananas	carrots	shark	60155	agree	1	No Category

**Example 5:** You can use the mutate function to change the variable type. The variable "total\_Income" is numeric. Change it to character.

`class(Illustration_Data$total_Income)`      Output: **numeric**

Code: `Illustration_Data <- Illustration_Data %>%  
 mutate(total_Income = as.character(total_Income))`

**changes the class of the column within the original dataframe itself.**

`class(Illustration_Data$total_Income)`      Output:  
**character**