Data Wrangling – Part 3

`mutate()` Function

`mutate()`: Creates a new column typically based on computations related to other columns in the dataset.

Command Illustration

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

For the illustration examples, assume the dataframe is the following:

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 will be a dataframe that looks like:

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	

We can use other functions (like `ifelse`) within mutate to help us make a new variable.

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

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

Output will be a dataframe that looks like:

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	

You can use the table function with two columns to count how many people fall into each category.

```
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
```

If you have multiple condition you can use the `case_when()` function and list out your possible options.

```
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"
  ))
```

Output will be a dataframe that looks like:

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 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"
  ))
```

Output will be a dataframe that looks like:

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 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:

Code:

```
class(Illustration_Data$total_Income)
```

Output: