# data_clean

2024-05-27

## what is a clean data

```
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

### Merge and any NA duplicated

```
df1 <- data.frame(ID = c(1, 2, 3, 4, 5),
                  length = c(5.1, 4.9, 4.7, 4.6, 5.0))
df2 <- data.frame(ID = c(1, 3, 5, 7, 9),
                  name = c("name1", "name2", "name3", "name4", "name5"))
data_merged <- merge(df1, df2, by = "ID")
print(data_merged)
```

```
##   ID length  name
## 1  1    5.1 name1
## 2  3    4.7 name2
## 3  5    5.0 name3
```

```
data_merged <- merge(df1, df2, by = "ID", all = TRUE)
print(data_merged)
```

```
##   ID length  name
## 1  1    5.1 name1
## 2  2    4.9  <NA>
## 3  3    4.7 name2
## 4  4    4.6  <NA>
## 5  5    5.0 name3
## 6  7     NA name4
## 7  9     NA name5
```

```
anyNA(data_merged)
```

```
## [1] TRUE
```

```
data_no_NA <- data_merged %>%
  na.omit()
anyNA(data_no_NA)
```

```
## [1] FALSE
```

```
anyDuplicated(data_no_NA)
```

```
## [1] 0
```

```
data_no_NA <- data_no_NA[!duplicated(data_no_NA),]
```

### Sum all data by some "day"

```
set.seed(123)
df <- data.frame(Week = sample(1:52, 500, replace = TRUE),
                 Hour = sample(1:24, 500, replace = TRUE),
                 Number = runif(500))

# Operation 1: Find the number of rows with a Week column value of 1
num_rows <- nrow(df[df$Week == 1, ])
cat("the number of rows with a Week column value of 1:", num_rows, "\n")
```

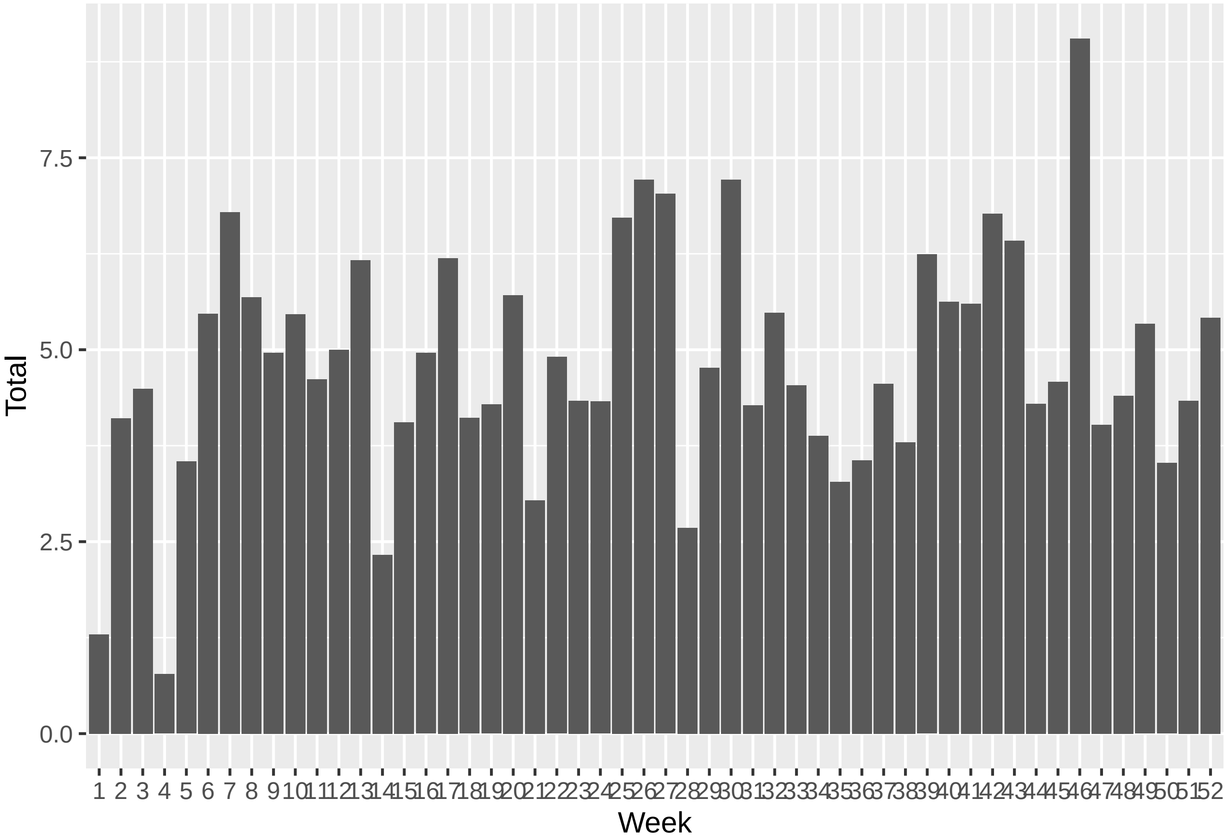
```
## the number of rows with a Week column value of 1: 2
```

```
# Operation 2: Summing numbers based on the same week value
df_sum <- df %>% group_by(Week) %>% summarise(Total = sum(Number))

# Operation 3: Sort the total numbers of each week in descending order and output the first row.
df_sorted <- df_sum %>% arrange(desc(Total))
top_row <- head(df_sorted, 1)
print(top_row)
```

```
## # A tibble: 1 x 2
##    Week Total
##   <int> <dbl>
## 1    46  9.05
```

```
df_sum$Week <- as.factor(df_sum$Week)
ggplot(data = df_sum,
       mapping = aes(x = Week, y = Total))+
  geom_col()
```


Total
Week