HW1

Himanshu Nimbarte

Installing Packages

Following commands are use to install packages

```
# First run this
if(!require('pacman'))
  install.packages("pacman")
```

```
Loading required package: pacman
```

```
library(pacman)

p_load(dlookr,
       DMwR2, # Data Mining with R functions
       GGally, # Pair-wise plots using ggplot2
       Hmisc, # Data analysis
       palmerpenguins, # Alternative to the Iris dataset
       tidyverse) # Data wrangling, manipulation, visualization
```

Loading Data

For loading data we will use function data()

```
data(algae, package = "DMwR2")

algae |> glimpse()
```

```
Rows: 200
Columns: 18
$ season <fct> winter, spring, autumn, spring, autumn, winter, summer, autumn,~
$ size   <fct> small, small, small, small, small, small, small, small, small, ~
$ speed  <fct> medium, medium, medium, medium, medium, high, high, high, mediu~
$ mxPH   <dbl> 8.00, 8.35, 8.10, 8.07, 8.06, 8.25, 8.15, 8.05, 8.70, 7.93, 7.7~
$ mnO2   <dbl> 9.8, 8.0, 11.4, 4.8, 9.0, 13.1, 10.3, 10.6, 3.4, 9.9, 10.2, 11.~
$ Cl     <dbl> 60.800, 57.750, 40.020, 77.364, 55.350, 65.750, 73.250, 59.067,~
$ NO3    <dbl> 6.238, 1.288, 5.330, 2.302, 10.416, 9.248, 1.535, 4.990, 0.886,~
$ NH4    <dbl> 578.000, 370.000, 346.667, 98.182, 233.700, 430.000, 110.000, 2~
$ oPO4   <dbl> 105.000, 428.750, 125.667, 61.182, 58.222, 18.250, 61.250, 44.6~
$ PO4    <dbl> 170.000, 558.750, 187.057, 138.700, 97.580, 56.667, 111.750, 77~
$ Chla   <dbl> 50.000, 1.300, 15.600, 1.400, 10.500, 28.400, 3.200, 6.900, 5.5~
$ a1     <dbl> 0.0, 1.4, 3.3, 3.1, 9.2, 15.1, 2.4, 18.2, 25.4, 17.0, 16.6, 32.~
$ a2     <dbl> 0.0, 7.6, 53.6, 41.0, 2.9, 14.6, 1.2, 1.6, 5.4, 0.0, 0.0, 0.0, ~
$ a3     <dbl> 0.0, 4.8, 1.9, 18.9, 7.5, 1.4, 3.2, 0.0, 2.5, 0.0, 0.0, 0.0, 2.~
$ a4     <dbl> 0.0, 1.9, 0.0, 0.0, 0.0, 0.0, 3.9, 0.0, 0.0, 2.9, 0.0, 0.0, 0.0~
$ a5     <dbl> 34.2, 6.7, 0.0, 1.4, 7.5, 22.5, 5.8, 5.5, 0.0, 0.0, 1.2, 0.0, 1~
$ a6     <dbl> 8.3, 0.0, 0.0, 0.0, 4.1, 12.6, 6.8, 8.7, 0.0, 0.0, 0.0, 0.0, 0.~
$ a7     <dbl> 0.0, 2.1, 9.7, 1.4, 1.0, 2.9, 0.0, 0.0, 0.0, 1.7, 6.0, 1.5, 2.1~
```

Central tendency: mean, median, mode

Mean

The **mean** function in R is used to calculate the arithmetic mean, also known as the average, of a numeric vector, matrix, or data frame. The mean is a measure of central tendency and represents the sum of all values divided by the number of values in the dataset.

```
algae$a1 |>
  mean()
```

```
[1] 16.9235
```

Median

In R, the **median** function is used to calculate the median of a numeric vector, matrix, or data frame. The median is a measure of central tendency that represents the middle value in a

dataset when it's sorted in ascending or descending order. If there is an even number of values, the median is the average of the two middle values.

```
algae$a1 |>
  median()
```

```
[1] 6.95
```

Mode

In R, the **mode** function is used to calculate the mode of a numeric vector or factor. The mode represents the value or values that occur most frequently in a dataset. Unlike the **mean** and **median**, which are measures of central tendency, the mode is a measure of the most common or frequent value(s).

```
Mode <- function(x, na.rm=FALSE){
if(na.rm) x<-x[!is.na(x)]
ux <- unique (x)
return (ux[which.max(tabulate(match(x, ux)))])
}

algae$a2 |> Mode()
```

```
[1] 0
```

`DMwRcentralValue()` function:

This function returns median or mode for the nominal variables.

```
# Numerical variable
algae$a1 |> centralValue()
```

```
[1] 6.95
```

```
# Nominal variable
algae$speed |> centralValue()
```

```
[1] "high"
```

Statistics of spread(variation)

Variance

In R, you can calculate the variance of a numeric vector, matrix, or data frame using the **var** function. Variance measures the spread or dispersion of data points around the mean. A higher variance indicates greater variability in the data.

```
algae$a1 |> var()
```

```
[1] 455.7532
```

Standard deviation

In R, you can calculate the standard deviation of a numeric vector, matrix, or data frame using the **sd** function. The standard deviation measures the dispersion or spread of data points around the mean. It's a common measure of variability in statistics.

```
algae$a1 |> sd()
```

```
[1] 21.34838
```

Range

In R, you can calculate the range of a numeric vector, which represents the difference between the maximum and minimum values in the dataset. To calculate the range, you can use the **range** function or simply subtract the minimum value from the maximum value.

```
algae$a1 |> range()
```

```
[1]  0.0 89.8
```

Maximum value

To find the maximum value in a numeric vector or matrix in R, you can use the **max** function. This function returns the highest value within the specified data.

```r
algae$a1 |> max()
```

```
[1] 89.8
```

Minimum value

To find the minimum value in a numeric vector or matrix in R, you can use the min function. This function returns the smallest value within the specified data.

```r
algae$a1 |> min()
```

```
[1] 0
```

Interquartile Range

The Interquartile Range (IQR) is a statistical measure that represents the spread or variability of a dataset. It is defined as the range between the first quartile (Q1) and the third quartile (Q3) of a dataset when it is sorted in ascending order.

3rd quartile (75%) - 1st quartile (25%)

```r
algae$a1 |> IQR()
```

```
[1] 23.3
```

Quantiles

Quantiles are values that divide a dataset into specified portions or intervals. Common quantiles include the median (which divides the data in half), quartiles (which divide the data into four equal parts), and percentiles (which divide the data into 100 equal parts).

```r
algae$a1 |> quantile()
```

```
   0%   25%   50%   75%  100% 
 0.00  1.50  6.95 24.80 89.80 
```

Specifying specific quantiles:

```
algae$a1 |> quantile(probs = c(0.2, 0.8))
```

```
  20%   80%
 1.20 32.18
```

Missing Values

```
library(purrr)
# Compute the total number of NA values in the dataset
nas <- algae %>%
  purrr::map_dbl(~sum(is.na(.))) %>%
  sum()

cat("The dataset contains ", nas, "NA values. \n")
```

```
The dataset contains  33 NA values.
```

```
# Compute the number of incomplete rows in the dataset
incomplete_rows <- algae %>%
  summarise_all(~!complete.cases(.)) %>%
  nrow()
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
i The deprecated feature was likely used in the dplyr package.
  Please report the issue at <https://github.com/tidyverse/dplyr/issues>.
```

```
cat("The dataset contains ", incomplete_rows, "(out of ", nrow(algae),") incomplete rows.
```

```
The dataset contains  200 (out of  200 ) incomplete rows.
```

Summaries of dataset

Base R's summary()

```
algae |> summary()
```

```
   season       size       speed         mxPH            mnO2       
autumn:40   large :45   high  :84   Min.   :5.600   Min.   : 1.500  
spring:53   medium:84   low   :33   1st Qu.:7.700   1st Qu.: 7.725  
summer:45   small :71   medium:83   Median :8.060   Median : 9.800  
winter:62                           Mean   :8.012   Mean   : 9.118  
                                    3rd Qu.:8.400   3rd Qu.:10.800  
                                    Max.   :9.700   Max.   :13.400  
                                    NA's   :1       NA's   :2       
      Cl                NO3              NH4                oPO4        
Min.   :  0.222   Min.   : 0.050   Min.   :    5.00   Min.   :  1.00  
1st Qu.: 10.981   1st Qu.: 1.296   1st Qu.:   38.33   1st Qu.: 15.70  
Median : 32.730   Median : 2.675   Median :  103.17   Median : 40.15  
Mean   : 43.636   Mean   : 3.282   Mean   :  501.30   Mean   : 73.59  
3rd Qu.: 57.824   3rd Qu.: 4.446   3rd Qu.:  226.95   3rd Qu.: 99.33  
Max.   :391.500   Max.   :45.650   Max.   :24064.00   Max.   :564.60  
NA's   :10        NA's   :2        NA's   :2          NA's   :2       
     PO4              Chla               a1              a2        
Min.   :  1.00   Min.   :  0.200   Min.   : 0.00   Min.   : 0.000  
1st Qu.: 41.38   1st Qu.:  2.000   1st Qu.: 1.50   1st Qu.: 0.000  
Median :103.29   Median :  5.475   Median : 6.95   Median : 3.000  
Mean   :137.88   Mean   : 13.971   Mean   :16.92   Mean   : 7.458  
3rd Qu.:213.75   3rd Qu.: 18.308   3rd Qu.:24.80   3rd Qu.:11.375  
Max.   :771.60   Max.   :110.456   Max.   :89.80   Max.   :72.600  
NA's   :2        NA's   :12                                        
      a3               a4               a5               a6        
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000  
Median : 1.550   Median : 0.000   Median : 1.900   Median : 0.000  
Mean   : 4.309   Mean   : 1.992   Mean   : 5.064   Mean   : 5.964  
3rd Qu.: 4.925   3rd Qu.: 2.400   3rd Qu.: 7.500   3rd Qu.: 6.925  
Max.   :42.800   Max.   :44.600   Max.   :44.400   Max.   :77.600  

      a7         
Min.   : 0.000  
1st Qu.: 0.000  
```

```
 Median : 1.000
 Mean   : 2.495
 3rd Qu.: 2.400
 Max.   :31.600
```

Hmisc's describe()

```
data("penguins")
penguins |> Hmisc::describe()
```

```
penguins

 8  Variables      344  Observations
--------------------------------------------------------------------------------
species
       n  missing distinct
     344        0        3

Value         Adelie Chinstrap    Gentoo
Frequency        152        68       124
Proportion     0.442     0.198     0.360
--------------------------------------------------------------------------------
island
       n  missing distinct
     344        0        3

Value         Biscoe     Dream Torgersen
Frequency        168       124        52
Proportion     0.488     0.360     0.151
--------------------------------------------------------------------------------
bill_length_mm
       n  missing distinct     Info     Mean      Gmd      .05      .10
     342        2      164        1    43.92    6.274    35.70    36.60
     .25      .50      .75      .90      .95
   39.23    44.45    48.50    50.80    51.99

lowest : 32.1 33.1 33.5 34   34.1, highest: 55.1 55.8 55.9 58   59.6
--------------------------------------------------------------------------------
bill_depth_mm
```

```
       n  missing distinct     Info     Mean      Gmd      .05      .10
     342        2       80        1    17.15    2.267     13.9     14.3
     .25      .50      .75      .90      .95
    15.6     17.3     18.7     19.5     20.0

lowest : 13.1 13.2 13.3 13.4 13.5, highest: 20.7 20.8 21.1 21.2 21.5
---------------------------------------------------------------------------
flipper_length_mm
       n  missing distinct     Info     Mean      Gmd      .05      .10
     342        2       55    0.999    200.9    16.03    181.0    185.0
     .25      .50      .75      .90      .95
   190.0    197.0    213.0    220.9    225.0

lowest : 172 174 176 178 179, highest: 226 228 229 230 231
---------------------------------------------------------------------------
body_mass_g
       n  missing distinct     Info     Mean      Gmd      .05      .10
     342        2       94        1     4202    911.8     3150     3300
     .25      .50      .75      .90      .95
    3550     4050     4750     5400     5650

lowest : 2700 2850 2900 2925 2975, highest: 5850 5950 6000 6050 6300
---------------------------------------------------------------------------
sex
       n  missing distinct
     333       11        2

Value      female   male
Frequency     165    168
Proportion  0.495  0.505
---------------------------------------------------------------------------
year
       n  missing distinct     Info     Mean      Gmd
     344        0        3    0.888     2008   0.8919

Value       2007  2008  2009
Frequency    110   114   120
Proportion 0.320 0.331 0.349

For the frequency table, variable is rounded to the nearest 0.02
---------------------------------------------------------------------------
```

dlookr's describe()

```
penguins |> dlookr::describe()
```

```
# A tibble: 5 x 26
  described_variables     n    na   mean      sd se_mean     IQR skewness
  <chr>               <int> <int>  <dbl>   <dbl>   <dbl>   <dbl>    <dbl>
1 bill_length_mm        342     2   43.9   5.46   0.295     9.27   0.0531
2 bill_depth_mm         342     2   17.2   1.97   0.107     3.1   -0.143
3 flipper_length_mm     342     2  201.   14.1    0.760    23      0.346
4 body_mass_g           342     2 4202.  802.    43.4    1200      0.470
5 year                  344     0 2008.    0.818  0.0441    2      -0.0537
# i 18 more variables: kurtosis <dbl>, p00 <dbl>, p01 <dbl>, p05 <dbl>,
#   p10 <dbl>, p20 <dbl>, p25 <dbl>, p30 <dbl>, p40 <dbl>, p50 <dbl>,
#   p60 <dbl>, p70 <dbl>, p75 <dbl>, p80 <dbl>, p90 <dbl>, p95 <dbl>,
#   p99 <dbl>, p100 <dbl>
```

Summaries on a subset of data

```
algae |>
  summarise(avgNO3 = mean(NO3, na.rm=TRUE),
            medA1 = median(a1))
```

```
# A tibble: 1 x 2
  avgNO3 medA1
   <dbl> <dbl>
1   3.28  6.95
```

```
algae |>
  select(mxPH:Cl) |>
  summarise_all(list(mean, median), na.rm = TRUE)
```

```
# A tibble: 1 x 6
  mxPH_fn1 mnO2_fn1 Cl_fn1 mxPH_fn2 mnO2_fn2 Cl_fn2
     <dbl>    <dbl>  <dbl>    <dbl>    <dbl>  <dbl>
1     8.01     9.12   43.6     8.06      9.8   32.7
```

```
algae |>
  select(a1:a7) |>
  summarise_all(funs(var))
```

```
Warning: `funs()` was deprecated in dplyr 0.8.0.
i Please use a list of either functions or lambdas:

# Simple named list: list(mean = mean, median = median)

# Auto named with `tibble::lst()`: tibble::lst(mean, median)

# Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
# A tibble: 1 x 7
     a1    a2    a3    a4    a5    a6    a7
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  456.  122.  48.3  19.5  56.1  136.  26.6
```

```
algae |>
  select(a1:a7) |>
  summarise_all(c("min", "max"))
```

```
# A tibble: 1 x 14
  a1_min a2_min a3_min a4_min a5_min a6_min a7_min a1_max a2_max a3_max a4_max
   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1      0      0      0      0      0      0      0   89.8   72.6   42.8   44.6
# i 3 more variables: a5_max <dbl>, a6_max <dbl>, a7_max <dbl>
```

Use sumarise() with group_by()

```
algae |>
  group_by(season, size) |>
  summarise(nObs = n(), mA7 = median(a7))
```

```
`summarise()` has grouped output by 'season'. You can override using the
`.groups` argument.
```

```
# A tibble: 12 x 4
# Groups:   season [4]
   season size    nObs   mA7
   <fct>  <fct>  <int> <dbl>
 1 autumn large     11  0
 2 autumn medium    16  1.05
 3 autumn small     13  0
 4 spring large     12  1.95
 5 spring medium    21  1
 6 spring small     20  0
 7 summer large     10  0
 8 summer medium    21  1
 9 summer small     14  1.45
10 winter large     12  0
11 winter medium    26  1.4
12 winter small     24  0
```

```
penguins |>
  group_by(species) |>
  summarise(var = var(bill_length_mm, na.rm = TRUE))
```

```
# A tibble: 3 x 2
  species     var
  <fct>     <dbl>
1 Adelie     7.09
2 Chinstrap 11.2
3 Gentoo     9.50
```

Aggregating Data

```
penguins |>
  group_by(species) |>
  reframe(var = quantile(bill_length_mm, na.rm = TRUE))
```

```
# A tibble: 15 x 2
   species     var
   <fct>     <dbl>
 1 Adelie     32.1
 2 Adelie     36.8
```

```
 3 Adelie     38.8
 4 Adelie     40.8
 5 Adelie     46
 6 Chinstrap  40.9
 7 Chinstrap  46.3
 8 Chinstrap  49.6
 9 Chinstrap  51.1
10 Chinstrap  58
11 Gentoo     40.9
12 Gentoo     45.3
13 Gentoo     47.3
14 Gentoo     49.6
15 Gentoo     59.6
```

```
penguins |>
  group_by(species) |>
  dlookr::describe(bill_length_mm)
```

```
# A tibble: 3 x 27
  described_variables species       n    na  mean    sd se_mean   IQR skewness
  <chr>               <fct>     <int> <int> <dbl> <dbl>   <dbl> <dbl>    <dbl>
1 bill_length_mm      Adelie      151     1  38.8  2.66   0.217  4      0.162
2 bill_length_mm      Chinstrap    68     0  48.8  3.34   0.405  4.73  -0.0906
3 bill_length_mm      Gentoo      123     1  47.5  3.08   0.278  4.25   0.651
# i 18 more variables: kurtosis <dbl>, p00 <dbl>, p01 <dbl>, p05 <dbl>,
#   p10 <dbl>, p20 <dbl>, p25 <dbl>, p30 <dbl>, p40 <dbl>, p50 <dbl>,
#   p60 <dbl>, p70 <dbl>, p75 <dbl>, p80 <dbl>, p90 <dbl>, p95 <dbl>,
#   p99 <dbl>, p100 <dbl>
```

Getting to know your data set

To load the dataset we will use data().

```
data("algae")
str(algae)
```

```
tibble [200 x 18] (S3: tbl_df/tbl/data.frame)
 $ season: Factor w/ 4 levels "autumn","spring",..: 4 2 1 2 1 4 3 1 4 4 ...
 $ size  : Factor w/ 3 levels "large","medium",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
 $ speed : Factor w/ 3 levels "high","low","medium": 3 3 3 3 3 1 1 1 3 1 ...
 $ mxPH  : num [1:200] 8 8.35 8.1 8.07 8.06 8.25 8.15 8.05 8.7 7.93 ...
 $ mnO2  : num [1:200] 9.8 8 11.4 4.8 9 13.1 10.3 10.6 3.4 9.9 ...
 $ Cl    : num [1:200] 60.8 57.8 40 77.4 55.4 ...
 $ NO3   : num [1:200] 6.24 1.29 5.33 2.3 10.42 ...
 $ NH4   : num [1:200] 578 370 346.7 98.2 233.7 ...
 $ oPO4  : num [1:200] 105 428.8 125.7 61.2 58.2 ...
 $ PO4   : num [1:200] 170 558.8 187.1 138.7 97.6 ...
 $ Chla  : num [1:200] 50 1.3 15.6 1.4 10.5 ...
 $ a1    : num [1:200] 0 1.4 3.3 3.1 9.2 15.1 2.4 18.2 25.4 17 ...
 $ a2    : num [1:200] 0 7.6 53.6 41 2.9 14.6 1.2 1.6 5.4 0 ...
 $ a3    : num [1:200] 0 4.8 1.9 18.9 7.5 1.4 3.2 0 2.5 0 ...
 $ a4    : num [1:200] 0 1.9 0 0 0 0 3.9 0 0 2.9 ...
 $ a5    : num [1:200] 34.2 6.7 0 1.4 7.5 22.5 5.8 5.5 0 0 ...
 $ a6    : num [1:200] 8.3 0 0 0 4.1 12.6 6.8 8.7 0 0 ...
 $ a7    : num [1:200] 0 2.1 9.7 1.4 1 2.9 0 0 0 1.7 ...
```

Here we will use install.packages() to install moments package which is use to use skewness function.

In R, there is a package called **moments** that provides functions for calculating various statistical moments, including mean, variance, skewness, and kurtosis. This package can be useful when you need to compute these moments for a dataset. To use the **moments** package, follow these steps:

```
install.packages("moments")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
library(moments)
```

```
Attaching package: 'moments'

The following objects are masked from 'package:dlookr':

    kurtosis, skewness
```

Calculating Skewness for particular attribute in the dataset

```
skewness(algae$a6)
```

```
[1] 3.160766
```

To check co-relation between the attributes we will use cor()

This will help us find co-relation between the attributes in the dataset

```
correlation_matrix <- cor(algae[, sapply(algae,is.numeric)])
print(correlation_matrix)
```

```
      mxPH mnO2 Cl NO3 NH4 oPO4 PO4 Chla          a1          a2          a3
mxPH     1   NA NA  NA  NA   NA  NA   NA          NA          NA          NA
mnO2    NA    1 NA  NA  NA   NA  NA   NA          NA          NA          NA
Cl      NA   NA  1  NA  NA   NA  NA   NA          NA          NA          NA
NO3     NA   NA NA   1  NA   NA  NA   NA          NA          NA          NA
NH4     NA   NA NA  NA   1   NA  NA   NA          NA          NA          NA
oPO4    NA   NA NA  NA  NA    1  NA   NA          NA          NA          NA
PO4     NA   NA NA  NA  NA   NA   1   NA          NA          NA          NA
Chla    NA   NA NA  NA  NA   NA  NA    1          NA          NA          NA
a1      NA   NA NA  NA  NA   NA  NA   NA  1.00000000 -0.29376781 -0.14656656
a2      NA   NA NA  NA  NA   NA  NA   NA -0.29376781  1.00000000  0.03214032
a3      NA   NA NA  NA  NA   NA  NA   NA -0.14656656  0.03214032  1.00000000
a4      NA   NA NA  NA  NA   NA  NA   NA -0.03795656 -0.17189273  0.01226780
a5      NA   NA NA  NA  NA   NA  NA   NA -0.29154923 -0.16046344 -0.10832262
a6      NA   NA NA  NA  NA   NA  NA   NA -0.27342831 -0.11641942 -0.17174646
a7      NA   NA NA  NA  NA   NA  NA   NA -0.21290633  0.04897715  0.05636698
               a4          a5          a6          a7
mxPH           NA          NA          NA          NA
mnO2           NA          NA          NA          NA
Cl             NA          NA          NA          NA
NO3            NA          NA          NA          NA
NH4            NA          NA          NA          NA
oPO4           NA          NA          NA          NA
PO4            NA          NA          NA          NA
Chla           NA          NA          NA          NA
a1   -0.03795656 -0.29154923 -0.27342831 -0.21290633
a2   -0.17189273 -0.16046344 -0.11641942  0.04897715
a3    0.01226780 -0.10832262 -0.17174646  0.05636698
a4    1.00000000 -0.10903079 -0.09009059  0.04306195
```

```
a5   -0.10903079  1.00000000  0.40473499 -0.02785471
a6   -0.09009059  0.40473499  1.00000000 -0.01349437
a7    0.04306195 -0.02785471 -0.01349437  1.00000000
```

To check for missing values we will use .na().

```
any_missing <- any(is.na(algae))
print(any_missing)
```

```
[1] TRUE
```

```
name = names(which(colSums(is.na(algae))>0))
print(name)
```

```
[1] "mxPH" "mnO2" "Cl"   "NO3"  "NH4"  "oPO4" "PO4"  "Chla"
```

```
algae[, c(name)]
```

```
# A tibble: 200 x 8
    mxPH  mnO2    Cl    NO3   NH4  oPO4   PO4  Chla
   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
 1  8      9.8  60.8  6.24  578   105   170   50
 2  8.35   8    57.8  1.29  370   429.  559.   1.3
 3  8.1   11.4  40.0  5.33  347.  126.  187.  15.6
 4  8.07   4.8  77.4  2.30   98.2  61.2 139.   1.4
 5  8.06   9    55.4 10.4   234.   58.2  97.6 10.5
 6  8.25  13.1  65.8  9.25  430    18.2  56.7 28.4
 7  8.15  10.3  73.2  1.54  110    61.2 112.   3.2
 8  8.05  10.6  59.1  4.99  206.   44.7  77.4  6.9
 9  8.7    3.4  22.0  0.886 103.   36.3  71    5.54
10  7.93   9.9   8    1.39    5.8  27.2  46.6  0.8
# i 190 more rows
```

In this case as there are numerical values missing we can use mean of the present values and can replace them with the missing values, otherwise we can also use median and mode.

We can omit the rows by using .omit() function

```r
algae1 <- algae #Duplicating dataframe
algae1_clean <- na.omit(algae1)
print(algae1_clean)
```

```
# A tibble: 184 x 18
   season size  speed   mxPH  mnO2    Cl    NO3   NH4  oPO4   PO4  Chla    a1
   <fct>  <fct> <fct>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 winter small medium  8      9.8  60.8  6.24  578   105   170   50      0
 2 spring small medium  8.35   8    57.8  1.29  370   429.  559.   1.3    1.4
 3 autumn small medium  8.1   11.4  40.0  5.33  347.  126.  187.  15.6    3.3
 4 spring small medium  8.07   4.8  77.4  2.30   98.2  61.2 139.   1.4    3.1
 5 autumn small medium  8.06   9    55.4 10.4   234.   58.2  97.6 10.5    9.2
 6 winter small high    8.25  13.1  65.8  9.25  430    18.2  56.7 28.4   15.1
 7 summer small high    8.15  10.3  73.2  1.54  110    61.2 112.   3.2    2.4
 8 autumn small high    8.05  10.6  59.1  4.99  206.   44.7  77.4  6.9   18.2
 9 winter small medium  8.7    3.4  22.0  0.886 103.   36.3  71    5.54  25.4
10 winter small high    7.93   9.9   8    1.39    5.8  27.2  46.6  0.8   17
# i 174 more rows
# i 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>,
#   a7 <dbl>
```