HW1

Shreya Kolte

GETTING TO KNOW R:

Installing required packages:

```
# First run this
install.packages("pacman")
```

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

```
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:

```
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~
```

To compute the central tendency: Mean, Median, Mode

Mean:

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

```
[1] 16.9235
```

Median:

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

```
[1] 6.95
```

Mode: There is no specific function for mode in R, so we will create a user defined function. But this method would only work for unimodal data.

```
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
```

Using the DMwR centralValue() function:

It will return the median for a numerical variable or the mode for nominal variables

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

```
[1] 6.95
```

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

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

Statistics of Spread (Variance):

Variance:

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

```
[1] 455.7532
```

Standard Deviation:

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

```
[1] 21.34838
```

Range: It gives us both maximum and minimum values

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

```
[1]  0.0 89.8
```

Maximum value:

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

```
[1] 89.8
```

Minimum value:

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

```
[1] 0
```

Interquartile Range:

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

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

```
[1] 23.3
```

Quantiles:

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

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

We can also specify 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 a 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
---------------------------------------------------------------------------
```

GMD is the mean absolute difference between any pairs of observations. A robust dispersion measure, especially for non-normally distributed data.

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:

dplyr's summarise() and summaries_all(), or use them with select() and group_by() to create summaries on subset of data.

Note: summarise() = semmarize()

```
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
```

summarise_all() can be used to apply any function that produces a scalar value to any column of a data frame table.

```
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>
```

Using summarise() 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:

This can be helpful for summary functions that don't return a scalar:

```
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
```

reframe() expects a scalar result by the function, but quantile returns a vector.

Using dlookr:

```
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>
```

EXERCISE:

Getting to know your dataset:

1. List datatypes of the attributes in your dataset: we use the str() function

```
```{r}
data("iris")
str(iris)
```
```

2. Check for skewness in data distribution in the attributes: we use the skewness() function

```
```{r}
skewness(iris$Sepal.Length)
```
```

3. Check for correlation among attributes: we use the cor() function

```
cor(iris[1:4], method="kendall")
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   1.00000000 -0.07699679    0.7185159   0.6553086
Sepal.Width   -0.07699679  1.00000000   -0.1859944  -0.1571257
Petal.Length   0.71851593 -0.18599442    1.0000000   0.8068907
Petal.Width    0.65530856 -0.15712566    0.8068907   1.0000000
```

4. Examine the extent of missing data. What would be the best way to deal with the missing data in this case?

```
data("airquality")
missing_val <- any(is.na(airquality)) #check missing values in complete dataset

missing_data <- sum(is.na(airquality$Ozone)) #check missing data in specific attribut

print(paste("Missing values in the data set:",missing_val), quote=FALSE)
```

```
[1] Missing values in the data set: TRUE
```

```
print(paste("Total number of missing values in the attribute of our dataset are:",mis
```

```
[1] Total number of missing values in the attribute of our dataset are: 37
```

```
print(paste("positions of our missing values are:"), quote=FALSE)
```

```
[1] positions of our missing values are:
```

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

```
[1] "Ozone"   "Solar.R"
```

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

```
   Ozone Solar.R
1     41     190
2     36     118
3     12     149
4     18     313
5     NA      NA
6     28      NA
7     23     299
8     19      99
9      8      19
10    NA     194
11     7      NA
12    16     256
13    11     290
14    14     274
15    18      65
16    14     334
17    34     307
18     6      78
19    30     322
20    11      44
21     1       8
22    11     320
23     4      25
24    32      92
25    NA      66
26    NA     266
27    NA      NA
28    23      13
```

29	45	252
30	115	223
31	37	279
32	NA	286
33	NA	287
34	NA	242
35	NA	186
36	NA	220
37	NA	264
38	29	127
39	NA	273
40	71	291
41	39	323
42	NA	259
43	NA	250
44	23	148
45	NA	332
46	NA	322
47	21	191
48	37	284
49	20	37
50	12	120
51	13	137
52	NA	150
53	NA	59
54	NA	91
55	NA	250
56	NA	135
57	NA	127
58	NA	47
59	NA	98
60	NA	31
61	NA	138
62	135	269
63	49	248
64	32	236
65	NA	101
66	64	175
67	40	314
68	77	276
69	97	267
70	97	272
71	85	175

72	NA	139
73	10	264
74	27	175
75	NA	291
76	7	48
77	48	260
78	35	274
79	61	285
80	79	187
81	63	220
82	16	7
83	NA	258
84	NA	295
85	80	294
86	108	223
87	20	81
88	52	82
89	82	213
90	50	275
91	64	253
92	59	254
93	39	83
94	9	24
95	16	77
96	78	NA
97	35	NA
98	66	NA
99	122	255
100	89	229
101	110	207
102	NA	222
103	NA	137
104	44	192
105	28	273
106	65	157
107	NA	64
108	22	71
109	59	51
110	23	115
111	31	244
112	44	190
113	21	259
114	9	36

```
115    NA     255
116    45     212
117   168     238
118    73     215
119    NA     153
120    76     203
121   118     225
122    84     237
123    85     188
124    96     167
125    78     197
126    73     183
127    91     189
128    47      95
129    32      92
130    20     252
131    23     220
132    21     230
133    24     259
134    44     236
135    21     259
136    28     238
137     9      24
138    13     112
139    46     237
140    18     224
141    13      27
142    24     238
143    16     201
144    13     238
145    23      14
146    36     139
147     7      49
148    14      20
149    30     193
150    NA     145
151    14     191
152    18     131
153    20     223
```

The above values are numeric and we can replace them with mean, median or mode of the columns or we can also omit the rows.