Lab 2: Data Structures

Complete all of the following questions, adding your inputs as code chunks (enclose within triple accent marks) within Rmarkdown.

The exercises are not marked and will not be factored into your course grade, but it is important to complete them to make sure you have the skills to answer assessment questions. You may consult any resource, including other students and the instructor. Please Knit this document to a PDF and upload your work via Canvas at the end of the session. Solutions will be posted for you to check your own answers.

String manipulation

1. Write your own sentence (at least 5 words) and store it in a string (character) variable.

```
Word<- 'We are in R lab Class and doing exercise in class.'
```

2. Print out the first 10 characters of the string to the screen (use `substr`).

```
substr(Word,1,10)
```

```
## [1] "We are in "
```

3. Print out the second character of the string.

```
substr(Word,0,2)
```

```
## [1] "We"
```

4. Print the sentence in ALL UPPERCASE (use `toupper`).

```
toupper(Word)
```

```
## [1] "WE ARE IN R LAB CLASS AND DOING EXERCISE IN CLASS."
```

5. Print out a string containing the sentence but with all spaces converted to underscores (use `gsub`).

```
gsub(' ','_',Word)
```

```
## [1] "We_are_in_R_lab_Class_and_doing_exercise_in_class."
```

6. Create a vector where each element of the vector contains a word from the string (use `strsplit`) and print it out.

```
v=c(strsplit(Word, ' '))
v
```

```
## [[1]]
##  [1] "We"       "are"      "in"       "R"        "lab"      "Class"   
##  [7] "and"      "doing"    "exercise" "in"       "class."
```

7. Print the words in your string in alphabetical order by storing the vector above in a variable, and then sorting it with `sort`.

```
Alph <- v[[1]]
sort(Alph)
```

```
##  [1] "and"      "are"      "Class"    "class."   "doing"    "exercise"
##  [7] "in"       "in"       "lab"      "R"        "We"
```

8. Create a variable containing the string "34 miles". (This is the approximate distance to Manchester.)

```
a= "34 miles"
a
```

```
## [1] "34 miles"
```

```
b='This is the approximate distance to Manchester'
paste(a,b)
```

```
## [1] "34 miles This is the approximate distance to Manchester"
```

9. Use `strsplit` and `as.numeric` to extract the numerical portion of the string, convert it to a number, and then divide it by two in order to calculate half the distance to Manchester in miles without any new data input. Save it as a variable and print it to the screen.

```
conv=strsplit(a, ' ')
str(conv)
```

```
## List of 1
##  $ : chr [1:2] "34" "miles"
```

```
dist=conv[[1]][1]
dist_half=as.numeric(dist)/2

dist_half
```

```
## [1] 17
```

10. Use the variable above to print out a single string that states: "Half the distance to Manchester is XX miles", except with the true value in place of XX. (Use `paste` to attach the strings.)

```
paste('Half the distance to Manchester is', dist_half,'muiles')
```

```
## [1] "Half the distance to Manchester is 17 muiles"
```

Matrices

11. Create a matrix (stored in variable m) of 3 rows and 5 columns, with every value equal to zero. Print out the matrix to confirm its shape.

```
m = matrix(0,3,5)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
```

```
dim(m)
```

```
## [1] 3 5
```

12. Set the value in the 2nd row, 3rd column to 1, then print out the matrix again.

```
m[2,3]=1
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    1    0    0
## [3,]    0    0    0    0    0
```

13. Set all the values in the 4th column to 2. Set all values in the 3rd row to 3. Print out the matrix again.

```
m[, 4]=2
m[3, ]=3

m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    2    0
## [2,]    0    0    1    2    0
## [3,]    3    3    3    3    3
```

14. Tack on a new row to the bottom of the matrix. The new row's elements should be all equal to 4. Then, tack on a new column to the right of the matrix. The new column's elements should be all equal to 5. (Use `rbind` and `cbind`.) Print out the matrix again and confirm that it now has 6 columns and 4 rows.

```r
m = rbind(m,c(4,4,4,4, 4))
m = cbind(m,c(5,5,5, 5))

m
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    0    0    2    0    5
## [2,]    0    0    1    2    0    5
## [3,]    3    3    3    3    3    5
## [4,]    4    4    4    4    4    5
```

```r
dim(m)
```

```
## [1] 4 6
```

15. Print the sum of all the elements in the matrix. (The result should be 60: if not, return to question 10).

```r
sum(m)
```

```
## [1] 60
```

16. Print a vector containing the sums of all the elements in each individual row (use `apply`)

```r
sum_v=apply(m,1,sum)

sum_v
```

```
## [1]  7  8 20 25
```

17. Print a vector containing the sums of all the elements in each individual column.

```r
sum_c_v=apply(m,2,sum)
sum_c_v
```

```
## [1]  7  7  8 11  7 20
```

18. Print a vector containing the means of all the elements in each individual column.

```r
mean_col=apply(m,2,mean)
mean_col
```

```
## [1] 1.75 1.75 2.00 2.75 1.75 5.00
```

19. Print a vector containing the medians of all the elements in each individual column.

```
median_col=apply(m,2,median)
median_col
```

```
## [1] 1.5 1.5 2.0 2.5 1.5 5.0
```

20. Print the transpose of the matrix (rows and columns swapped.)

```
t(m)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    3    4
## [2,]    0    0    3    4
## [3,]    0    1    3    4
## [4,]    2    2    3    4
## [5,]    0    0    3    4
## [6,]    5    5    5    5
```

Named vectors

21. Suppose you collected the following data on your friends' birthdays:

    Alice: 31/8
    Bob: 5/5
    Carol: 11/12
    David: 17/10

    Create a string vector containing *just* the birthdays.

```
birthdays= c('31/8',' 5/5','11/12','17/10')
birthdays
```

```
## [1] "31/8"  " 5/5"  "11/12" "17/10"
```

22. Add names to all four elements in your vector using your friends' names above.

```
names(birthdays) = c('Alice','Bob','Carol','David')
birthdays
```

```
##   Alice     Bob   Carol   David
##  "31/8"  " 5/5" "11/12" "17/10"
```

23. Print out Carol's birthday using her name as a subscript.

```
birthdays['Carol']
```

```
##   Carol
## "11/12"
```

Lists

Suppose you have some miscellaneous data about a car: It has 4 seats. Its top speed is 100 miles per hour. The make is Hyundai. The model is Elantra. The transmission is standard. The tyre pressure readings in PSI are 30, 31, 29, 33.

24. Create a list containing all this data in a single variable. Print out information about its structure.

```
l_24=list(product='car',
          seats=4,
          speed_mh=100,
          model='Elantra',
          transmission='standard',
          tyre_pressure=c(30, 31, 29, 33) )

l_24
```

```
## $product
## [1] "car"
##
## $seats
## [1] 4
##
## $speed_mh
## [1] 100
##
## $model
## [1] "Elantra"
##
## $transmission
## [1] "standard"
##
## $tyre_pressure
## [1] 30 31 29 33
```

```
str(l_24)
```

```
## List of 6
##  $ product      : chr "car"
##  $ seats        : num 4
##  $ speed_mh     : num 100
##  $ model        : chr "Elantra"
##  $ transmission : chr "standard"
##  $ tyre_pressure: num [1:4] 30 31 29 33
```

25. Print out the mean tyre pressure (using an operation on your list variable).

```
mean(l_24$tyre_pressure)
```

```
## [1] 30.75
```

Data Frames

26. Reload the student summary table from the in-class exercise. How many students are represented in the survey, and how many variables are present?

```
stu_survey = read.csv('survey.csv',as.is=FALSE)
stu_survey
```

```
##    age height gender   football_club     beverage siblings av_height_est dogs
## 1   35     70   male         Chelsea         <NA>        4            60   No
## 2   26     75   male         Chelsea       coffee        3            66  Yes
## 3   25     73   male       Liverpool       coffee        1            60  Yes
## 4   23     69   male         Arsenal       coffee        1            70  Yes
## 5   23     69   male       Barcelona       coffee        1            NA  Yes
## 6   21     72   male       Liverpool       coffee        2            66  Yes
## 7   NA     67 female           none          none        0            68  Yes
## 8   22     69   male       Liverpool       coffee        2            68  Yes
## 9   NA     64 female           none           tea        5            64   No
## 10  35     71   male Manchester City       coffee        0            70  Yes
## 11  24     65 female     Real Madrid       coffee        2            66  Yes
## 12  22     68 female            <NA>       coffee        3            67  Yes
## 13  22     66   male     Real Madrid       coffee        1            70  Yes
## 14  24     67   male            <NA>       coffee        4            59  Yes
## 15  NA     NA   male            <NA> energy drink        0            68  Yes
## 16  24     67   male    Wisla Krakow         cola        1            67  Yes
## 17  25     61 female       Liverpool       coffee        1            70  Yes
## 18  24     71   male         Chelsea       coffee        2            65  Yes
## 19  22     64 female       Liverpool       coffee        1            68  Yes
## 20  21     69 female             PSG          tea        2            65  Yes
## 21  23     66   male            <NA>          tea        2            70  Yes
## 22  25     59 female       Liverpool          tea        1            70  Yes
## 23  24     NA female            <NA>       coffee       NA            NA  Yes
## 24  28     72   male         Chelsea         cola        2            68   No
## 25  29     65 female            <NA>       coffee        1            65  Yes
## 26  21     69   male       Liverpool         cola        1            65  Yes
## 27  25     65 female            <NA>       coffee        1            70   No
## 28  22     70   male           Leeds       coffee        1            69  Yes
## 29  33     60 female            none          tea        1            71   No
## 30  30     66 female            none       coffee        3            60   No
## 31  33     68   male       Liverpool energy drink        3            67   No
## 32  25     74   male            <NA>         <NA>        3            60  Yes
##    hometown_pop berlin_dist_est berlin_dist_unc distance fave_nums
## 1        200000           10000            5000      5.0      7,14
## 2       2000000             100              80      4.0      8,14
## 3      14000000             200              50      0.0      7,12
## 4         20000            3000             500      0.7       7,5
## 5          1000            1000             500      2.0     8,420
## 6         60000            1500             250     10.0    17,257
## 7         38000             800             100      3.0   2,4,6,8
## 8        500000             800             200      4.0      7,22
## 9        504000              NA              NA      0.8       9,7
## 10      4000000            3000             500      0.7      3,33
## 11      3000000             600             200      1.0     4,6,8
## 12    500000000              NA              NA      0.4       3,2
## 13           NA            1000              NA      1.0       7,4
## 14        25000            1500             500      1.5     21,64
## 15        60000            1000              NA      1.5     10,46
## 16         1337             495              40      0.0     13,21
## 17      3000000            4000              NA      1.0       5,2
## 18        10000             600             200      0.5       6,8
## 19        44000              NA              NA      8.0      6,21
## 20          300             300             100      4.0      2,19
## 21         1408            1000              50      5.0       3,5
## 22      3300000             500             100      0.6      7,20
## 23       400000            2000             102      2.0       6,8
## 24      1000000            3222            1000     15.0      2,22
## 25      1080100             700             600      2.5      9,11
## 26         6000             750              40      1.5    28,196
## 27      3000000             600             100      1.0      26,8
## 28        21000             900             300      1.0      27,e
## 29        95000            1000              70      8.0       3,7
## 30          500            2000             500      2.0      92,6
## 31       165000             910              NA      3.0       3,8
## 32        69000            3400             200      3.0       3,7
##    wake_time_wkday wake_time_wkend colour1 colour2
## 1            06:00           07:00    Blue   White
## 2            06:00           07:00    Blue   Black
## 3            09:00           11:30    Blue   Black
## 4            07:00           08:00     Red   White
## 5            07:00           08:30     Red   Black
## 6            08:00           12:00   Green   Black
## 7            07:00           07:00    Blue   Black
## 8            08:00           08:00     Red   Black
## 9            09:00           10:00     Red   White
## 10           07:00           08:00    Blue   Black
## 11           09:00           11:00     Red   White
## 12           07:30           08:30    Blue   Black
## 13           07:00           06:00     Red   Black
## 14           08:00           11:00    Blue   Black
## 15           05:30           05:30     Red   Black
## 16           07:30           09:00   Green   White
## 17           08:00           10:00     Red   Black
## 18           07:30           08:30   Green   Black
## 19           07:30           12:00     Red   Black
## 20           10:00           13:00    Blue   White
## 21           07:00           07:00    Blue   White
## 22           07:00           09:00    Blue   Black
## 23           06:30           09:00    Blue   Black
## 24           05:00           05:00    Blue   Black
## 25           07:30           08:30   Green   Black
## 26           08:00           06:45     Red   Black
## 27           08:00           10:00     Red   Black
## 28           08:00           08:00   Green   White
## 29           06:30           07:30    <NA>   Black
## 30           05:30           07:00    Blue   Black
## 31           06:30           05:00   Green   White
## 32           06:00           05:00    Blue   Black
```

27. Generate a summary of the dataframe using the `summary` command. Then, answer the following questions:

  a. What is the mean and median height of the students?
  b. What is the most popular football club?
  c. What is the most popular type of caffeinated beverage?

  You can simply read them off the summary table and enter them down below the code chunk by hand. (In principle you can also calculate them using various R functions, but this is less straightforward.)

```r
summary(stu_survey)
```

```
##       age            height         gender       football_club
##  Min.   :21.00   Min.   :59.00   female:13   Liverpool  :8
##  1st Qu.:22.00   1st Qu.:65.25   male  :19   Chelsea    :4
##  Median :24.00   Median :68.00               none       :2
##  Mean   :25.38   Mean   :67.70               none       :2
##  3rd Qu.:26.00   3rd Qu.:70.00               Real Madrid:2
##  Max.   :35.00   Max.   :75.00               (Other)    :6
##  NA's   :3       NA's   :2                   NA's       :8
##          beverage     siblings     av_height_est    dogs     hometown_pop
##  coffee      :19   Min.   :0.000   Min.   :59.00   No : 7   Min.   :      300
##  cola        : 3   1st Qu.:1.000   1st Qu.:65.00   Yes:25   1st Qu.:    20500
##  energy drink: 2   Median :1.000   Median :67.00            Median :    95000
##  none        : 1   Mean   :1.774   Mean   :66.40            Mean   : 17309730
##  tea         : 5   3rd Qu.:2.500   3rd Qu.:69.75            3rd Qu.:  1540050
##  NA's        : 2   Max.   :5.000   Max.   :71.00            Max.   :500000000
##                    NA's   :1       NA's   :2                NA's   :1
##  berlin_dist_est berlin_dist_unc     distance        fave_nums  wake_time_wkday
##  Min.   :  100   Min.   :  40.0   Min.   : 0.000   3,7    : 2   07:00  :7
##  1st Qu.:  600   1st Qu.: 100.0   1st Qu.: 0.950   6,8    : 2   08:00  :7
##  Median : 1000   Median : 200.0   Median : 1.750   10,46  : 1   07:30  :5
##  Mean   : 1616   Mean   : 451.3   Mean   : 2.928   13,21  : 1   06:00  :3
##  3rd Qu.: 2000   3rd Qu.: 500.0   3rd Qu.: 4.000   17,257 : 1   06:30  :3
##  Max.   :10000   Max.   :5000.0   Max.   :15.000   2,19   : 1   09:00  :3
##  NA's   :3       NA's   :7                         (Other):24   (Other):4
##  wake_time_wkend  colour1    colour2
##  07:00   : 5      Blue :14   Black:23
##  08:00   : 4      Green: 6   White: 9
##  08:30   : 4      Red  :11
##  05:00   : 3      NA's : 1
##  09:00   : 3
##  10:00   : 3
##  (Other):10
```

```r
#Mean   :67.70
#Liverpool  :8
#coffee      :19

is.na(stu_survey$height)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
mean(stu_survey$height, na.rm=TRUE)
```

```
## [1] 67.7
```

28. Using a logical subscript, print out the distance from Liverpool city centre of everyone who listed Liverpool as their top football club.

```
stu_survey$distance[stu_survey$football_club== 'Liverpool']
```

```
##  [1]  0.0 10.0  4.0   NA   NA   NA  1.0  8.0   NA  0.6   NA   NA  1.5   NA  3.0
## [16]   NA
```

29. Print out the distances of everyone who listed a football club other than Liverpool as their top football club. (This should *not* include those with no preference / no answer!)

```
na.omit(stu_survey$distance[stu_survey$football_club != 'Liverpool'])
```

```
##  [1]  5.0  4.0  0.7  2.0  3.0  0.8  0.7  1.0  1.0  0.0  0.5  4.0 15.0  1.0  8.0
## [16]  2.0
## attr(,"na.action")
## [1]  9 11 12 16 17 19 20 24
## attr(,"class")
## [1] "omit"
```

30. Calculate the average height of women in the survey, and the average height of men in the survey.

```
round(tapply(stu_survey$height,stu_survey$gender,mean, na.rm=TRUE))
```

```
## female   male
##     64     70
```

31. Calculate the mean heights of everyone in the survey broken down by their preferred beverage (i.e.: the mean heights of coffee-drinkers, tea-drinkers, etc.) Note: You can do this with a single command in R.

```
round(tapply(stu_survey$height,stu_survey$beverage,mean, na.rm=TRUE))
```

```
##       coffee          cola energy drink         none          tea
##           68            69           68           67           64
```

32. Produce a histogram of (all) student heights. Note: be careful about the default break points!

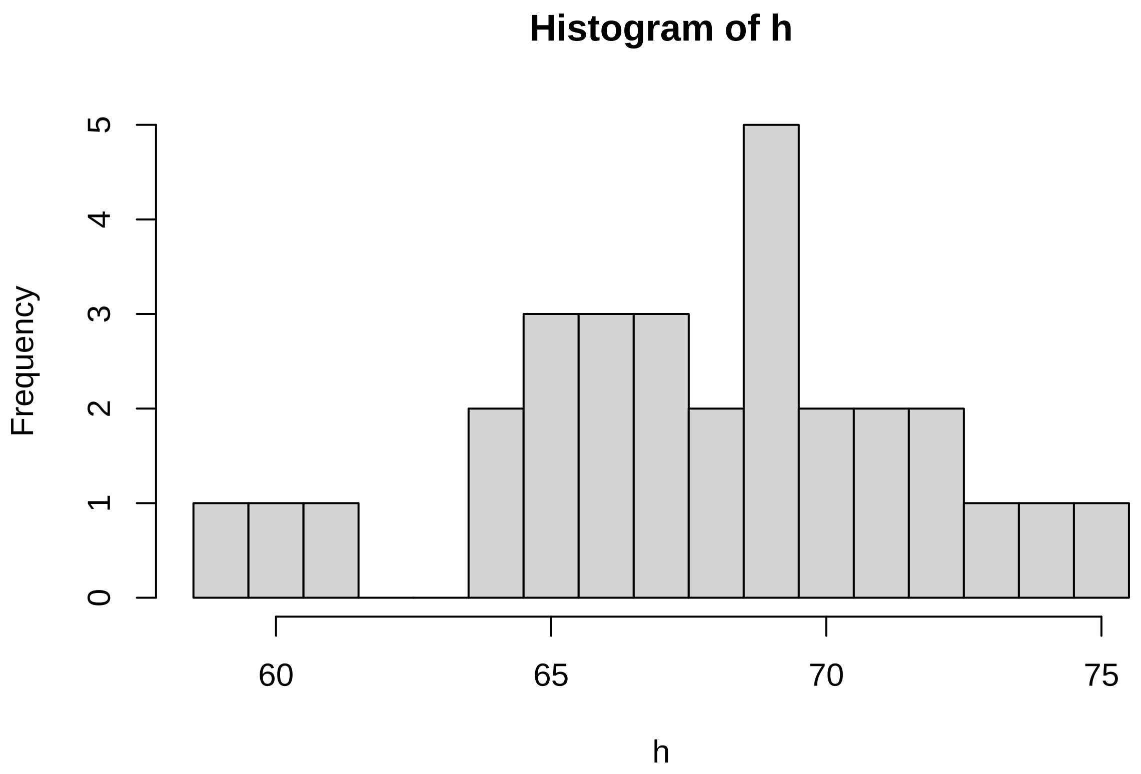
```
h= na.omit(stu_survey$height)
min(h)
```

```
## [1] 59
```

```
max(h)
```

```
## [1] 75
```

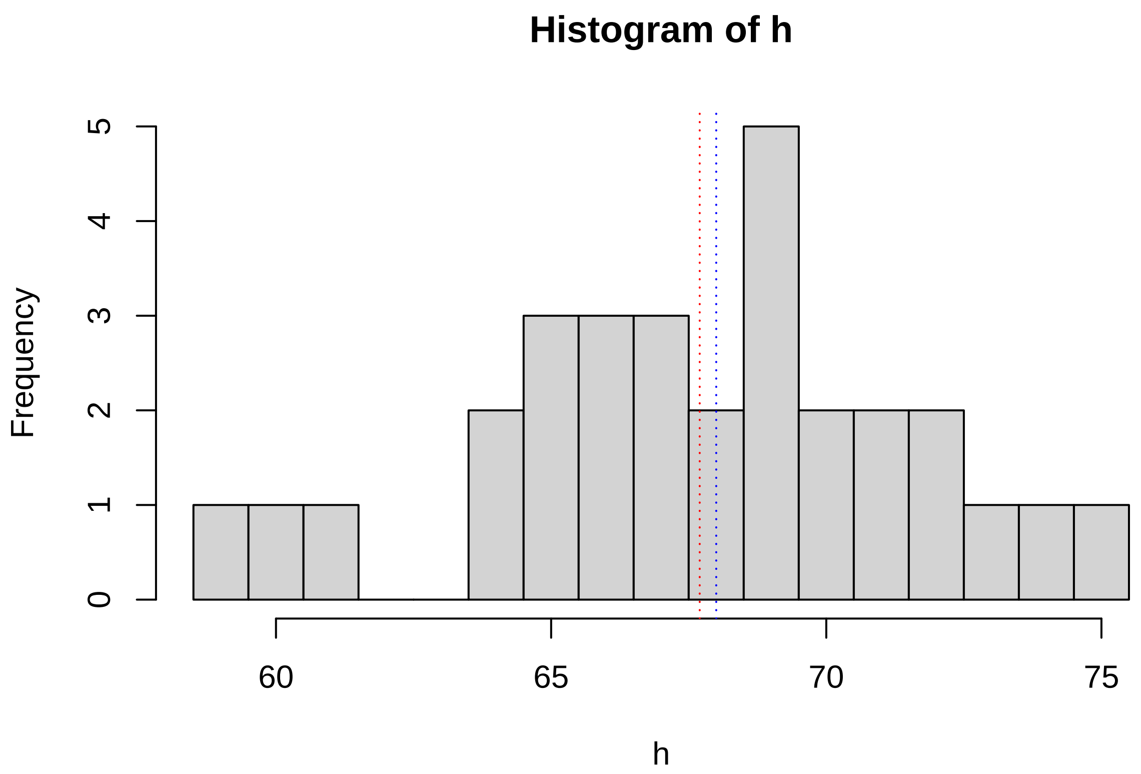
```
hist(h, breaks=seq(58.5, 75.5) )
```



33. Overplot (plot on top of the previous plot) two vertical lines of different colours: one at the MEAN value, and one at the MEDIAN value. *(Note: in Rmarkdown you'll need to repeat the plotting command from #31 in your code chunk for #32.)*

```
hist(h, breaks=seq(58.5, 75.5) )
abline( v=mean(h),
       col='red',
       lty=3,
      lwd=1)

abline( v=median(h),
       col='blue',
       lty=3,
      lwd=1)
```



34. Suppose that one of the students who did not specify a football club preference decides to support a team. Choose a student that did not specify a football club (or specified 'none') and update the dataframe by assigning a football team of your choice to that student. Then print out an updated frequency table containing the number of students supporting each football club.

```
which_noclub = which(stu_survey$football_club=='none')
stu_survey$football_club[which_noclub[1]] = 'Liverpool'
table(stu_survey$football_club)
```

```
##
##         Arsenal       Barcelona         Chelsea           Leeds       Liverpool
##               1               1               4               1               9
## Manchester City            none           none              PSG     Real Madrid
##               1               1               2               1               2
##    Wisla Krakow
##               1
```

Larger Data Frames and Normality

The file abalone.csv (available on Canvas) contains data from a sample of blacklip abalones (*Haliotis rubra*) gathered from the Bass Straits of Tasmania in the early 1990s. The measurements include various dimensions of the shell (length, diameter, height) and weight after various levels of treatment. "Rings" gives the number of ring layers inside the shell and is a proxy for age since about one ring layer is added per year. Dimensions are in metres and weights are in kilograms.

35. Download the file abalone.csv to your local computer. Load it into R as a data frame, and print out the first few lines using `head`.

```
data=read.csv('abalone.csv')
head(data)
```

```
##   sex length diameter height weight shucked.weight viscera.weight shell.weight
## 1   M  0.455    0.365  0.095 0.5140         0.2245         0.1010        0.150
## 2   M  0.350    0.265  0.090 0.2255         0.0995         0.0485        0.070
## 3   F  0.530    0.420  0.135 0.6770         0.2565         0.1415        0.210
## 4   M  0.440    0.365  0.125 0.5160         0.2155         0.1140        0.155
## 5   I  0.330    0.255  0.080 0.2050         0.0895         0.0395        0.055
## 6   I  0.425    0.300  0.095 0.3515         0.1410         0.0775        0.120
##   rings
## 1    15
## 2     7
## 3     9
## 4    10
## 5     7
## 6     8
```

36. How many different variables (columns) are in the table, what are their names, and what sort of data do they contain? (Continuous, categorical, others)? How many measurements are there in the sample? (Use the `str` command.)

```
str(data)
```

```
## 'data.frame':    4177 obs. of  9 variables:
##  $ sex           : chr  "M" "M" "F" "M" ...
##  $ length        : num  0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ diameter      : num  0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ height        : num  0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ weight        : num  0.514 0.226 0.677 0.516 0.205 ...
##  $ shucked.weight: num  0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ viscera.weight: num  0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ shell.weight  : num  0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ rings         : int  15 7 9 10 7 8 20 16 9 19 ...
```

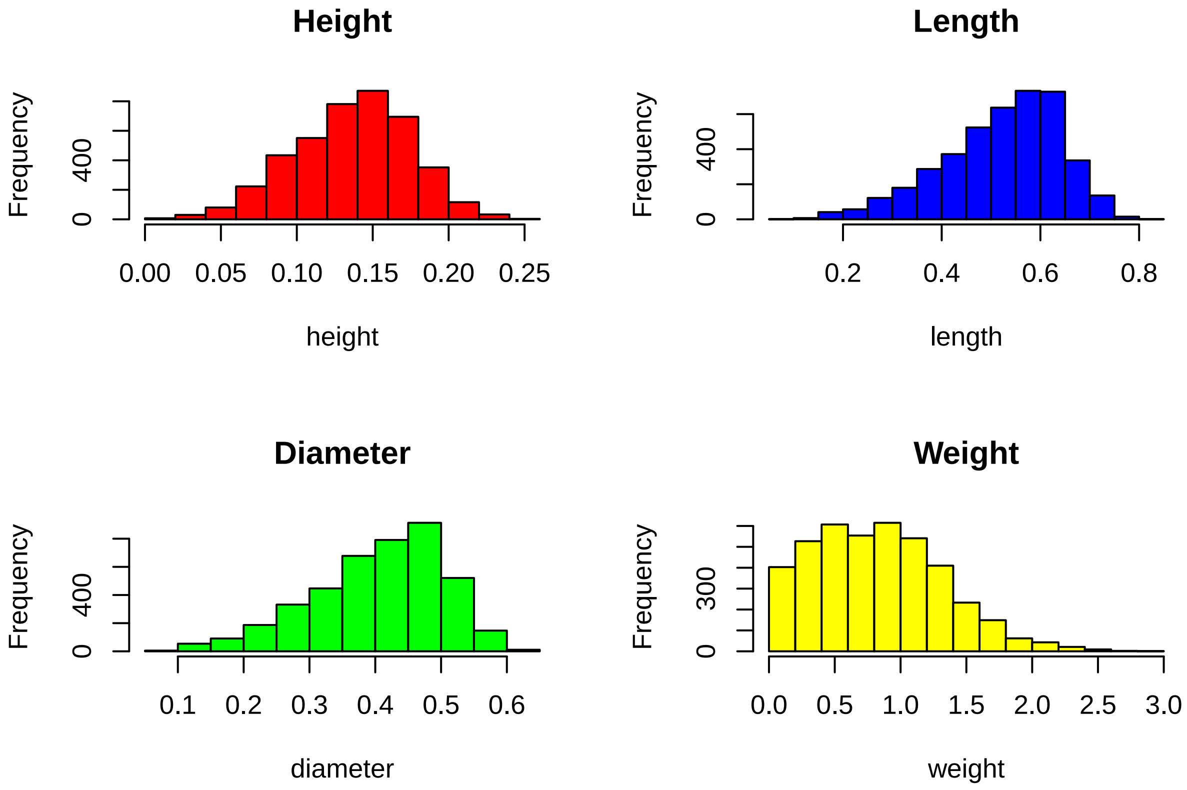
37. Produce a summary table of the abalone dataset using `summary`. Make sure you understand what all the numbers mean.

```
summary(data)
```

```
##      sex                length         diameter          height      
##  Length:4177        Min.   :0.075   Min.   :0.0550   Min.   :0.0000  
##  Class :character   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150  
##  Mode  :character   Median :0.545   Median :0.4250   Median :0.1400  
##                     Mean   :0.524   Mean   :0.4079   Mean   :0.1392  
##                      3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650
##                      Max.   :0.815   Max.   :0.6500   Max.   :0.2500
##                                                       NA’s   :1
##      weight       shucked.weight   viscera.weight    shell.weight
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.0005   Min.   :0.0015
##  1st Qu.:0.4415   1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300
##  Median :0.7995   Median :0.3360   Median :0.1710   Median :0.2340
##  Mean   :0.8287   Mean   :0.3594   Mean   :0.1806   Mean   :0.2388
##  3rd Qu.:1.1530   3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290
##  Max.   :2.8255   Max.   :1.4880   Max.   :0.7600   Max.   :1.0050
##
##      rings
##  Min.   : 1.000
##  1st Qu.: 8.000
##  Median : 9.000
##  Mean   : 9.934
##  3rd Qu.:11.000
##  Max.   :29.000
##
```

38. Produce a four-panel set of plots (use `par(mfrow)` to set this up) containing histograms of the following four quantities: height, length, diameter, weight. Make the axis labels and titles reader-friendly and use colours to fill the histogram bars.

```
par(mfrow=c(2,2))
hist(data$height,col='red',
     main='Height',
     xlab='height')
hist(data$length,
     col='blue',
     main='Length',
     xlab='length')
hist(data$diameter,
     col='green',
     main='Diameter',
     xlab='diameter')
hist(data$weight,
     col='yellow',
     main='Weight',
     xlab='weight')
```

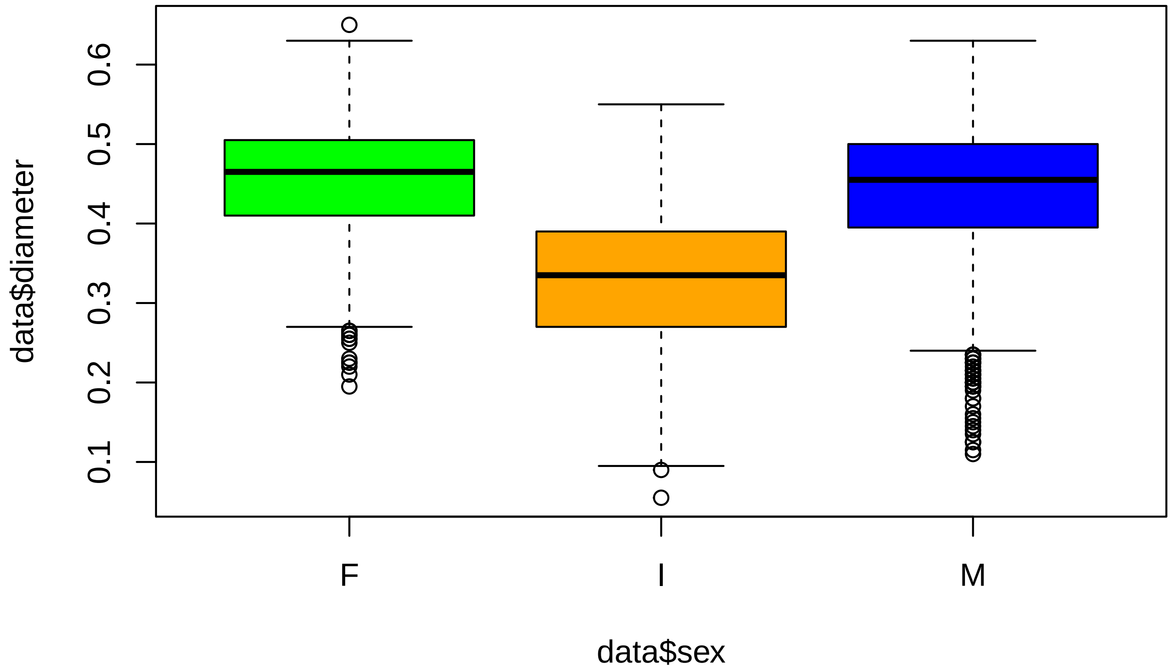


39. Do any of these quantities appear to be normally distributed?

Write your response here. weight, diameter and length does not look like normally distributed

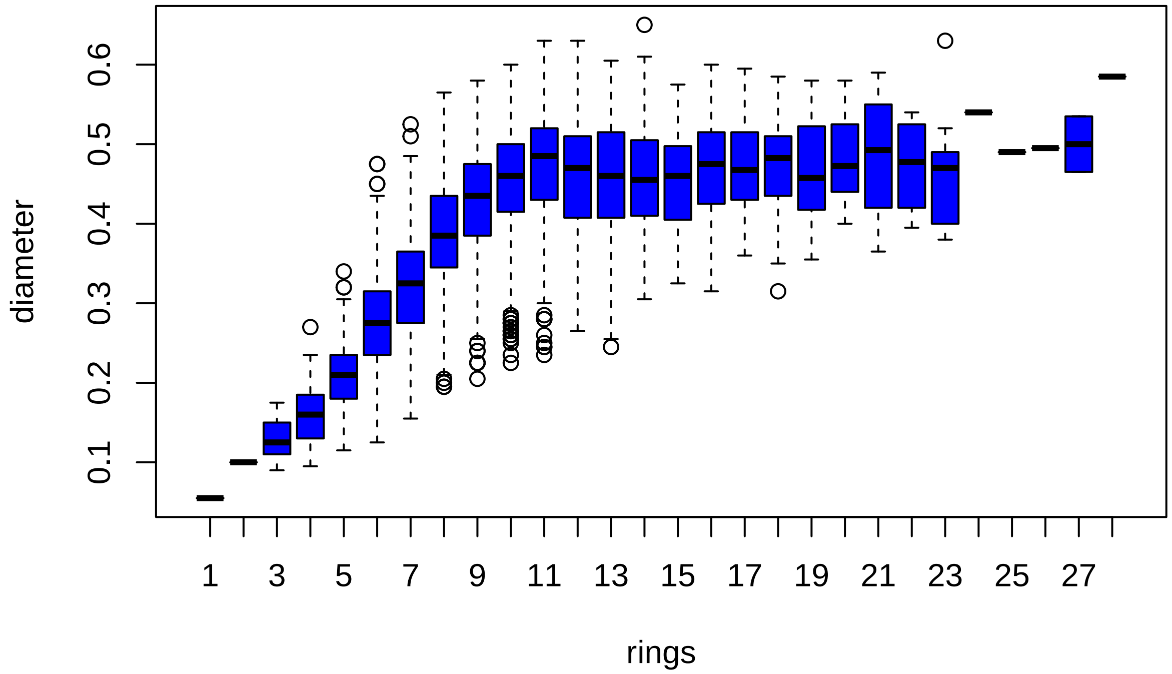
40. Produce a (one-panel) box-and-whisker plot summarising the distribution of abalone diameter for the three sex categories (F, I, or M). Be sure to label the axes. Do the distributions of females and males seem obviously different? What about for "I", meaning indeterminate/unknown sex?

```
boxplot(data$diameter~ data$sex,
        col=c('green','orange','blue'))
```



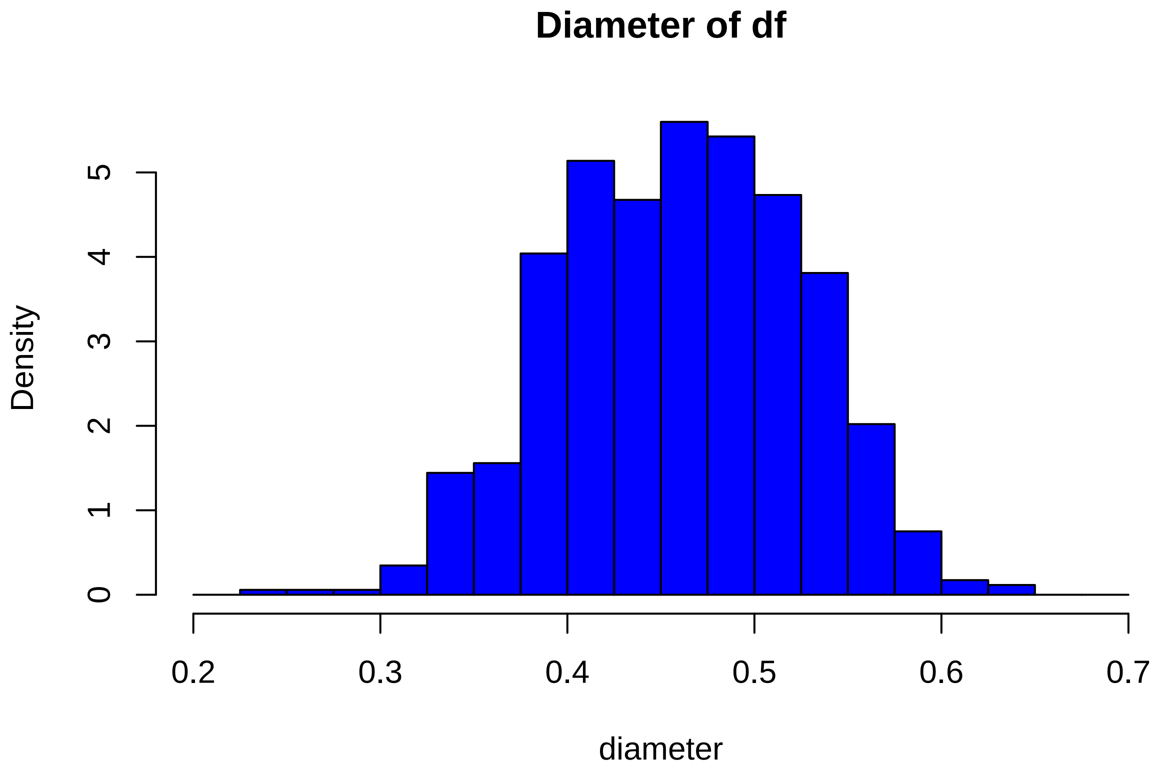
41. Produce a box-and-whisker plot showing the distribution of abalone diameter as a function of number of rings (a measurement proxy for age in years.)

```
boxplot(data$diameter~data$rings,
        col='blue',
        xlab='rings',
        ylab='diameter')
```



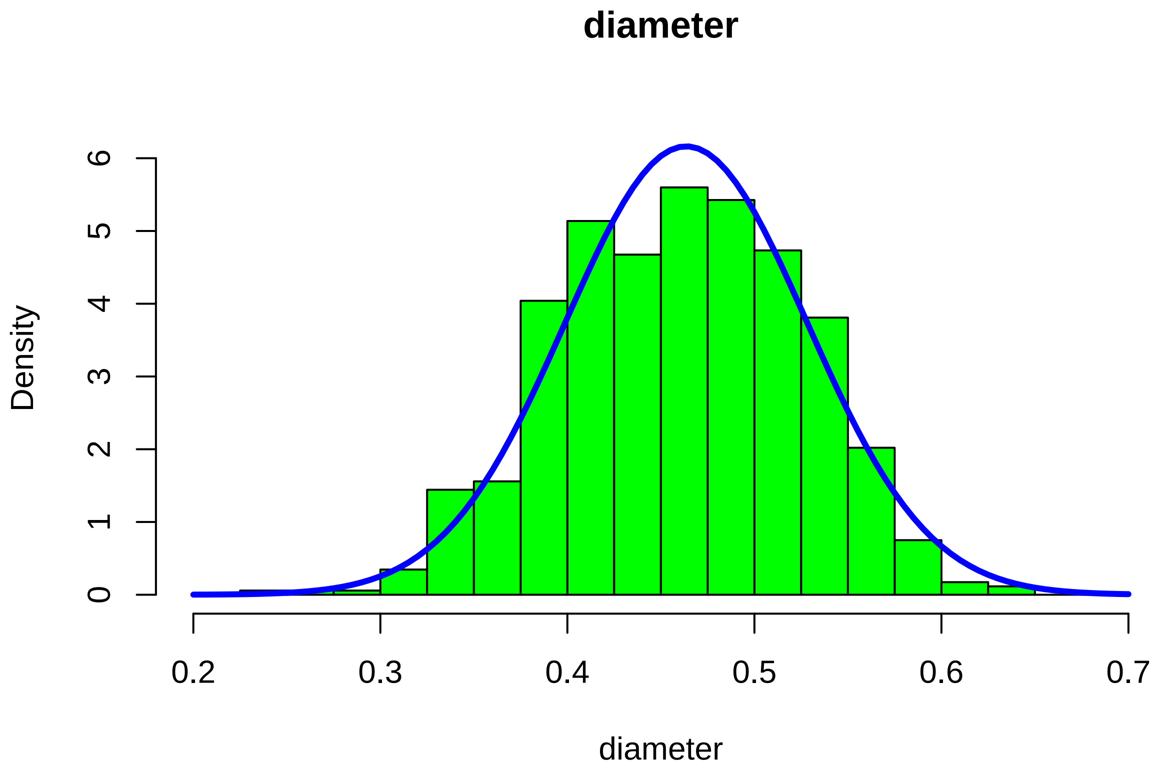
42. Create a new dataframe that contains a subset of the original data frame: specifically, it includes only those rows for which nrings is greater than or equal to 13. Then, produce a histogram of the diameters of this subset. (Set `freq=FALSE` to plot probability density instead of frequency, and you may want to specify the breaks to get better resolution.) Do these look normally distributed?

```
df = data[data$rings >= 13,]
hist(df$diameter,col='blue',
     main='Diameter of df',
     xlab='diameter',
     breaks=seq(0.2,0.7,0.025),
     freq=FALSE)
```



43. Check your assessment of normality by over-plotting a normal distribution on top of the histogram above, using the appropriate μ and σ parameters to describe the population. (Start by defining a plotting variable along the x-axis using `seq`, and then calculate the normal PDF as a function of this variable using the function `dnorm`.)

```
hist(df$diameter,col='green',main='diameter',
     xlab='diameter',
     breaks=seq(0.2,0.7,0.025),
     freq=FALSE,ylim=c(0,6.5))
d = seq(0.2,0.7,0.005)
m = mean(df$diameter)
s = sd(df$diameter)
lines(d,dnorm(d,mean=m,sd=s),col='blue',lwd=3)
```



44. Explain why the central limit theorem did not seem to hold for the entire data set, even though it did for a subset.

Write your response here. if we repeate it different sample result will turned into normal distribution