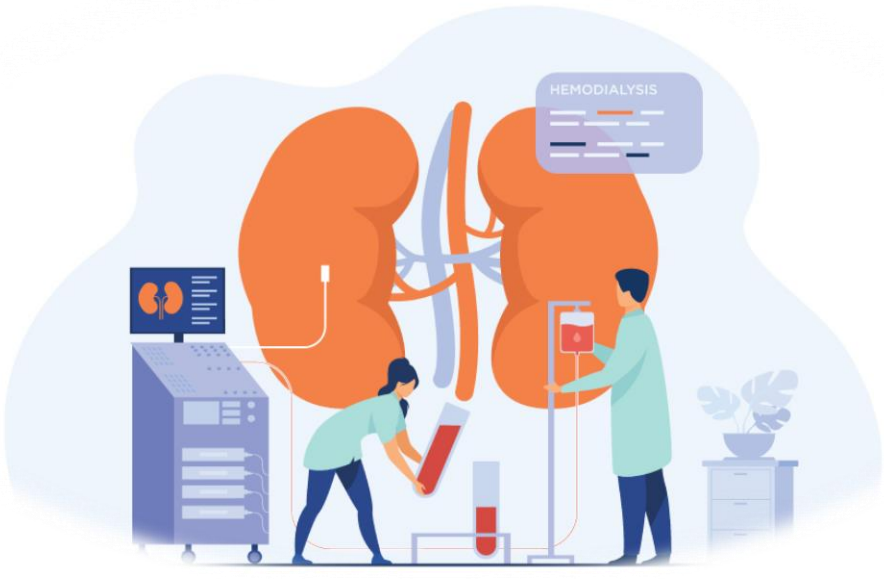
FACULTY OF COMPUTING & INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

FCIT
KAU

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز

CPIS-490 Final Project

Student Name	Student ID
Arwa Saad Alqahtani	1805926
Sondus Alshaghdali	1807275
Layali Alkhamisi	1806384



Task 1: Discovery

1. Read/Load the dataset into R.

```
> data <- read.csv(file.path("c:", "kidney_disease.csv"))
> data
    id age  bp    sg al su      rbc        pc         pcc         ba bgr  bu   sc   sod pot hemo pcv    wc  rc htn   dm cad appet  pe ane classification
1    0  48  80 1.020  1  0           normal notpresent notpresent 121  36  1.2    NA  NA 15.4  44  7800 5.2 yes  yes  no  good  no  no            ckd
2    1   7  50 1.020  4  0           normal notpresent notpresent  NA  18  0.8    NA  NA 11.3  38  6000      no  no  no good  no  no            ckd
3    2  62  80 1.010  2  3   normal  normal notpresent notpresent 423  53  1.8    NA  NA  9.6  31  7500      no  yes  no  poor  no yes            ckd
4    3  48  70 1.005  4  0   normal abnormal    present notpresent 117  56  3.8 111.0 2.5 11.2  32  6700 3.9 yes  no  no  poor yes yes            ckd
5    4  51  80 1.010  2  0   normal  normal notpresent notpresent 106  26  1.4    NA  NA 11.6  35  7300 4.6  no  no  no  good  no  no            ckd
6    5  60  90 1.015  3  0           notpresent notpresent  74  25  1.1 142.0 3.2 12.2  39  7800 4.4 yes  yes  no  good yes  no            ckd
7    6  68  70 1.010  0  0           normal notpresent notpresent 100  54 24.0 104.0 4.0 12.4  36           no  no  no  good  no  no            ckd
8    7  24  NA 1.015  2  4   normal abnormal notpresent notpresent 410  31  1.1    NA  NA 12.4  44  6900   5  no  yes  no  good yes  no            ckd
9    8  52 100 1.015  3  0   normal abnormal    present notpresent 138  60  1.9    NA  NA 10.8  33  9600 4.0 yes  yes  no  good  no yes            ckd
10   9  53  90 1.020  2  0 abnormal abnormal    present notpresent  70 107  7.2 114.0 3.7  9.5  29 12100 3.7 yes  yes  no  poor  no yes            ckd
11  10  50  60 1.010  2  4           abnormal    present notpresent 490  55  4.0    NA  NA  9.4  28           yes  yes  no  good  no yes            ckd
12  11  63  70 1.010  3  0 abnormal abnormal    present notpresent 380  60  2.7 131.0 4.2 10.8  32  4500 3.8 yes  yes  no  poor yes  no            ckd
13  12  68  70 1.015  3  1           normal    present notpresent 208  72  2.1 138.0 5.8  9.7  28 12200 3.4 yes  yes yes  poor yes  no            ckd
14  13  68  70    NA NA NA                   notpresent notpresent  98  86  4.6 135.0 3.4  9.8           yes  yes yes  poor yes  no            ckd
15  14  68  80 1.010  3  2   normal abnormal    present    present 157  90  4.1 130.0 6.4  5.6  16 11000 2.6 yes  yes yes  poor yes  no            ckd
16  15  40  80 1.015  3  0           normal notpresent notpresent  76 162  9.6 141.0 4.9  7.6  24  3800 2.8 yes  no  no  good  no yes            ckd
17  16  47  70 1.015  2  0           normal notpresent notpresent  99  46  2.2 138.0 4.1 12.6           no  no  no  good  no  no            ckd
18  17  47  80    NA NA NA                   notpresent notpresent 114  87  5.2 139.0 3.7 12.1           yes  no  no  poor  no  no            ckd
19  18  60 100 1.025  0  3           normal notpresent notpresent 263  27  1.3 135.0 4.3 12.7  37 11400 4.3 yes  yes yes  good  no  no            ckd
20  19  62  60 1.015  1  0         abnormal    present notpresent 100  31  1.6    NA  NA 10.3  30  5300 3.7 yes  no yes  good  no  no            ckd
21  20  61  80 1.015  2  0 abnormal abnormal notpresent notpresent 173 148  3.9 135.0 5.2  7.7  24  9200 3.2 yes  yes yes  poor yes yes            ckd
22  21  60  90    NA NA NA                   notpresent notpresent  NA 180 76.0   4.5  NA 10.9  32  6200 3.6 yes  yes yes  good  no  no            ckd
23  22  48  80 1.025  4  0   normal abnormal notpresent notpresent  95 163  7.7 136.0 3.8  9.8  32  6900 3.4 yes  no  no  good  no yes            ckd
24  23  21  70 1.010  0  0           normal notpresent notpresent  NA  NA   NA    NA  NA   NA           no  no  no  poor  no yes            ckd
25  24  42 100 1.015  4  0   normal abnormal notpresent    present  NA  50  1.4 129.0 4.0 11.1  39  8300 4.6 yes  no  no  poor  no  no            ckd
26  25  61  60 1.025  0  0           normal notpresent notpresent 108  75  1.9 141.0 5.2  9.9  29  8400 3.7 yes  yes no  good  no yes            ckd
27  26  75  80 1.015  0  0           normal notpresent notpresent 156  45  2.4 140.0 3.4 11.6  35 10300   4 yes  yes no  poor  no  no            ckd
28  27  69  70 1.010  3  4   normal abnormal notpresent notpresent 264  87  2.7 130.0 4.0 12.5  37  9600 4.1 yes  yes yes  good yes  no            ckd
29  28  75  70    NA  1  3           notpresent notpresent 123  31  1.4    NA  NA   NA           no  yes  no  good  no  no            ckd
30  29  68  70 1.005  1  0 abnormal abnormal    present notpresent  NA  28  1.4    NA  NA 12.9  38           no   no yes  good  no  no            ckd
31  30  NA  70    NA NA NA                   notpresent notpresent  93 155  7.3 132.0 4.9   NA           yes  yes  no  good  no  no            ckd
32  31  73  90 1.015  3  0         abnormal    present notpresent 107  33  1.5 141.0 4.6 10.1  30  7800   4  no   no  no  poor  no  no            ckd
33  32  61  90 1.010  1  1           normal notpresent notpresent 159  39  1.5 133.0 4.9 11.3  34  9600 4.0 yes  yes no  poor  no  no            ckd
34  33  60 100 1.020  2  0 abnormal abnormal notpresent notpresent 140  55  2.5    NA  NA 10.1  29           yes  no  no  poor  no  no            ckd
35  34  70  70 1.010  1  0   normal    present    present 171 153  5.2    NA  NA   NA           no  yes  no  poor  no  no            ckd
36  35  65  90 1.020  2  1 abnormal  normal notpresent notpresent 270  39  2.0    NA  NA 12.0  36  9800 4.9 yes  yes  no  poor  no yes            ckd
37  36  76  70 1.015  1  0   normal  normal notpresent notpresent  92  29  1.8 133.0 3.9 10.3  32           yes   no  no  good  no  no            ckd
38  37  72  80    NA NA NA                   notpresent notpresent 137  65  3.4 141.0 4.7  9.7  28  6900 2.5 yes  yes  no  poor  no yes          ckd\t
 [ reached 'max' / getOption("max.print") -- omitted 362 rows ]
>
```

2. In preparation of our data, we Use all the attributes except albumin and pus cell

By using select= -c(al,pc) , the "-" sign indicates dropping variables

```
> #remove alumin and pus cell attributes, The '-' sign indicates dropping variables
> new_data=subset(data, select=-c(al,pc))
> new_data
    id age  bp    sg su      rbc         pcc         ba bgr  bu   sc   sod pot hemo pcv    wc  rc htn   dm cad appet  pe ane classification
1    0  48  80 1.020  0           notpresent notpresent 121  36  1.2    NA  NA 15.4  44  7800 5.2 yes  yes  no  good  no  no            ckd
2    1   7  50 1.020  0           notpresent notpresent  NA  18  0.8    NA  NA 11.3  38  6000      no  no  no  good  no  no            ckd
3    2  62  80 1.010  3   normal  notpresent notpresent 423  53  1.8    NA  NA  9.6  31  7500      no  yes  no  poor  no yes            ckd
4    3  48  70 1.005  0   normal     present notpresent 117  56  3.8 111.0 2.5 11.2  32  6700 3.9 yes  no  no  poor yes yes            ckd
5    4  51  80 1.010  0   normal  notpresent notpresent 106  26  1.4    NA  NA 11.6  35  7300 4.6  no  no  no  good  no  no            ckd
6    5  60  90 1.015  0           notpresent notpresent  74  25  1.1 142.0 3.2 12.2  39  7800 4.4 yes  yes  no  good yes  no            ckd
7    6  68  70 1.010  0           notpresent notpresent 100  54 24.0 104.0 4.0 12.4  36           no  no  no  good  no  no            ckd
8    7  24  NA 1.015  4   normal  notpresent notpresent 410  31  1.1    NA  NA 12.4  44  6900   5  no  yes  no  good yes  no            ckd
9    8  52 100 1.015  0   normal     present notpresent 138  60  1.9    NA  NA 10.8  33  9600 4.0 yes  yes  no  good  no yes            ckd
10   9  53  90 1.020  0 abnormal     present notpresent  70 107  7.2 114.0 3.7  9.5  29 12100 3.7 yes  yes  no  poor  no yes            ckd
11  10  50  60 1.010  4              present notpresent 490  55  4.0    NA  NA  9.4  28           yes  yes  no  good  no yes            ckd
12  11  63  70 1.010  0 abnormal     present notpresent 380  60  2.7 131.0 4.2 10.8  32  4500 3.8 yes  yes  no  poor yes  no            ckd
13  12  68  70 1.015  1              present notpresent 208  72  2.1 138.0 5.8  9.7  28 12200 3.4 yes  yes yes  poor yes  no            ckd
14  13  68  70    NA NA           notpresent notpresent  98  86  4.6 135.0 3.4  9.8           yes  yes yes  poor yes  no            ckd
15  14  68  80 1.010  2   normal     present    present 157  90  4.1 130.0 6.4  5.6  16 11000 2.6 yes  yes yes  poor yes  no            ckd
16  15  40  80 1.015  0           notpresent notpresent  76 162  9.6 141.0 4.9  7.6  24  3800 2.8 yes  no  no  good  no yes            ckd
17  16  47  70 1.015  0           notpresent notpresent  99  46  2.2 138.0 4.1 12.6           no  no  no  good  no  no            ckd
18  17  47  80    NA NA           notpresent notpresent 114  87  5.2 139.0 3.7 12.1           yes  no  no  poor  no  no            ckd
19  18  60 100 1.025  3           notpresent notpresent 263  27  1.3 135.0 4.3 12.7  37 11400 4.3 yes  yes yes  good  no  no            ckd
20  19  62  60 1.015  0              present notpresent 100  31  1.6    NA  NA 10.3  30  5300 3.7 yes  no yes  good  no  no            ckd
21  20  61  80 1.015  0 abnormal  notpresent notpresent 173 148  3.9 135.0 5.2  7.7  24  9200 3.2 yes  yes yes  poor yes yes            ckd
22  21  60  90    NA NA           notpresent notpresent  NA 180 76.0   4.5  NA 10.9  32  6200 3.6 yes  yes yes  good  no  no            ckd
23  22  48  80 1.025  0   normal  notpresent notpresent  95 163  7.7 136.0 3.8  9.8  32  6900 3.4 yes  no  no  good  no yes            ckd
24  23  21  70 1.010  0           notpresent notpresent  NA  NA   NA    NA  NA   NA           no  no  no  poor  no yes            ckd
25  24  42 100 1.015  0   normal  notpresent    present  NA  50  1.4 129.0 4.0 11.1  39  8300 4.6 yes  no  no  poor  no  no            ckd
26  25  61  60 1.025  0           notpresent notpresent 108  75  1.9 141.0 5.2  9.9  29  8400 3.7 yes  yes no  good  no yes            ckd
27  26  75  80 1.015  0           notpresent notpresent 156  45  2.4 140.0 3.4 11.6  35 10300   4 yes  yes no  poor  no  no            ckd
28  27  69  70 1.010  4   normal  notpresent notpresent 264  87  2.7 130.0 4.0 12.5  37  9600 4.1 yes  yes yes  good yes  no            ckd
29  28  75  70    NA  3           notpresent notpresent 123  31  1.4    NA  NA   NA           no  yes  no  good  no  no            ckd
30  29  68  70 1.005  0 abnormal     present notpresent  NA  28  1.4    NA  NA 12.9  38           no   no yes  good  no  no            ckd
31  30  NA  70    NA NA           notpresent notpresent  93 155  7.3 132.0 4.9   NA           yes  yes  no  good  no  no            ckd
32  31  73  90 1.015  0              present notpresent 107  33  1.5 141.0 4.6 10.1  30  7800   4  no   no  no  poor  no  no            ckd
33  32  61  90 1.010  1           notpresent notpresent 159  39  1.5 133.0 4.9 11.3  34  9600 4.0 yes  yes no  poor  no  no            ckd
34  33  60 100 1.020  0 abnormal  notpresent notpresent 140  55  2.5    NA  NA 10.1  29           yes  no  no  poor  no  no            ckd
35  34  70  70 1.010  0   normal     present    present 171 153  5.2    NA  NA   NA           no  yes  no  poor  no  no            ckd
36  35  65  90 1.020  1 abnormal  notpresent notpresent 270  39  2.0    NA  NA 12.0  36  9800 4.9 yes  yes  no  poor  no yes            ckd
37  36  76  70 1.015  0   normal  notpresent notpresent  92  29  1.8 133.0 3.9 10.3  32           yes   no  no  good  no  no            ckd
38  37  72  80    NA NA           notpresent notpresent 137  65  3.4 141.0 4.7  9.7  28  6900 2.5 yes  yes  no  poor  no yes          ckd\t
39  38  69  80 1.020  0 abnormal  notpresent notpresent  NA 103  4.1 132.0 5.9 12.5                yes   no  no  good  no  no            ckd
40  39  82  80 1.010  2   normal  notpresent notpresent 140  70  3.4 136.0 4.2 13.0  40  9800 4.2 yes  yes  no  good  no  no            ckd
```

3. **Check the structure of the data (# of instances, attributes, datatypes, missing values, etc.)**

```
# 3-Check the structure of the data

# numbers of instances & attributes
dim(new_data)

#check the datatypes
str(new_data)

#number of missing values
# returns a vector (F F F T)
is.na(new_data)
#sum of missing values
sum(is.na(new_data))

# 4-Provide general statistical description

summary(new_data)
```

-Number of instances: 400-

-Number of attribute: 24

```
> # numbers of rows & columns
> dim(new_data)
[1] 400  24
```

- the most data type in our dataset is: CHR

```
> #check the datatypes
> str(new_data)
'data.frame':   400 obs. of  24 variables:
 $ id             : int  0 1 2 3 4 5 6 7 8 9 ...
 $ age            : num  48 7 62 48 51 60 68 24 52 53 ...
 $ bp             : num  80 50 80 70 80 90 70 NA 100 90 ...
 $ sg             : num  1.02 1.02 1.01 1 1.01 ...
 $ su             : num  0 0 3 0 0 0 0 4 0 0 ...
 $ rbc            : chr  "" "" "normal" "normal" ...
 $ pcc            : chr  "notpresent" "notpresent" "notpresent" "present" ...
 $ ba             : chr  "notpresent" "notpresent" "notpresent" "notpresent" ...
 $ bgr            : num  121 NA 423 117 106 74 100 410 138 70 ...
 $ bu             : num  36 18 53 56 26 25 54 31 60 107 ...
 $ sc             : num  1.2 0.8 1.8 3.8 1.4 1.1 24 1.1 1.9 7.2 ...
 $ sod            : num  NA NA NA 111 NA 142 104 NA NA 114 ...
 $ pot            : num  NA NA NA 2.5 NA 3.2 4 NA NA 3.7 ...
 $ hemo           : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
 $ pcv            : chr  "44" "38" "31" "32" ...
 $ wc             : chr  "7800" "6000" "7500" "6700" ...
 $ rc             : chr  "5.2" "" "" "3.9" ...
 $ htn            : chr  "yes" "no" "no" "yes" ...
 $ dm             : chr  "yes" "no" "yes" "no" ...
 $ cad            : chr  "no" "no" "no" "no" ...
 $ appet          : chr  "good" "good" "poor" "poor" ...
 $ pe             : chr  "no" "no" "no" "yes" ...
 $ ane            : chr  "no" "no" "yes" "yes" ...
 $ classification: chr  "ckd" "ckd" "ckd" "ckd" ...
```

-the number of missing values: 424

```
> #number of missing values
> # returns a vector (F F F T)
> is.na(new_data)
      id   age    bp    sg    su   rbc   pcc    ba   bgr    bu    sc   sod   pot  hemo   pcv    wc    rc   htn    dm   cad appet    pe   ane classification
1  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
2  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
3  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
4  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
5  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
6  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
7  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
8  FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
9  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
10 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
11 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
12 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
13 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
14 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
15 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
16 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
17 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
18 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
19 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
20 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
21 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
22 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
23 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
24 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
25 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
26 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
27 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
28 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
29 FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
30 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
31 FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
32 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
33 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
34 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
35 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
36 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
37 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
38 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
39 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
40 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
41 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE          FALSE
 [ reached getOption("max.print") -- omitted 359 rows ]
> sum(is.na(new_data))
[1] 424
>
```

4. Provide statistical description.

We used summary() function to display statistical description of the data which is Min, 1st quartile, Median, Mean, 3rd quartile, and Max
for example, the minimum age is 2 years old, and the maximum age is 90 years old

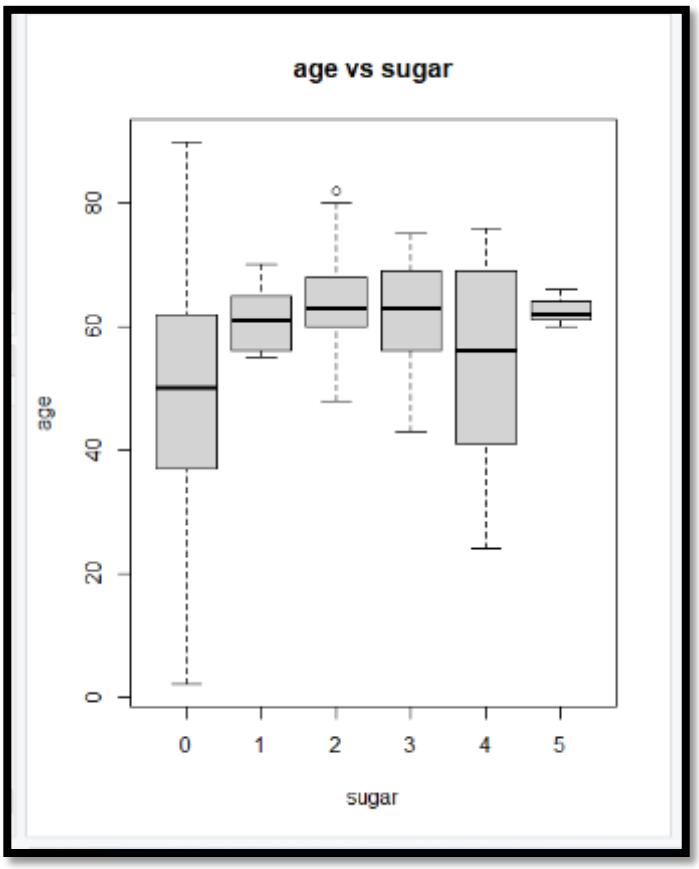
```
> summary(new_data)
       id              age              bp              sg              su              rbc                pcc                ba                bgr
 Min.   :  0.00   Min.   : 2.00   Min.   : 50.00   Min.   :1.005   Min.   :0.0000   Length:400         Length:400         Length:400         Min.   : 22
 1st Qu.: 99.75   1st Qu.:42.00   1st Qu.: 70.00   1st Qu.:1.010   1st Qu.:0.0000   Class :character   Class :character   Class :character   1st Qu.: 99
 Median :199.50   Median :55.00   Median : 80.00   Median :1.020   Median :0.0000   Mode  :character   Mode  :character   Mode  :character   Median :121
 Mean   :199.50   Mean   :51.48   Mean   : 76.47   Mean   :1.017   Mean   :0.4501                                                         Mean   :148
 3rd Qu.:299.25   3rd Qu.:64.50   3rd Qu.: 80.00   3rd Qu.:1.020   3rd Qu.:0.0000                                                         3rd Qu.:163
 Max.   :399.00   Max.   :90.00   Max.   :180.00   Max.   :1.025   Max.   :5.0000                                                         Max.   :490
                  NA's   :9       NA's   :12       NA's   :47      NA's   :49                                                             NA's   :44
       bu               sc              sod              pot              hemo             pcv                wc                rc                htn
 Min.   :  1.50   Min.   : 0.400   Min.   :  4.5   Min.   : 2.500   Min.   : 3.10   Length:400         Length:400         Length:400         Length:400
 1st Qu.: 27.00   1st Qu.: 0.900   1st Qu.:135.0   1st Qu.: 3.800   1st Qu.:10.30   Class :character   Class :character   Class :character   Class :character
 Median : 42.00   Median : 1.300   Median :138.0   Median : 4.400   Median :12.65   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   : 57.43   Mean   : 3.072   Mean   :137.5   Mean   : 4.627   Mean   :12.53
 3rd Qu.: 66.00   3rd Qu.: 2.800   3rd Qu.:142.0   3rd Qu.: 4.900   3rd Qu.:15.00
 Max.   :391.00   Max.   :76.000   Max.   :163.0   Max.   :47.000   Max.   :17.80
 NA's   :19       NA's   :17       NA's   :87      NA's   :88       NA's   :52
      dm                cad               appet                pe                ane            classification
 Length:400         Length:400         Length:400         Length:400         Length:400         Length:400
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

5. **Visualize the data/selected variables in three different ways. Describe the plots in detail.**

- **Boxplot**

The sugar levels range from 0 to 5.
in the first box plot (level 0) the minimum value is 2 year, the median is 50 year and the maximum is 90 year.
In the (level 1) the minimum value is 57 years, the median is 60 year and the maximum value is 69 year.
In the (level 2) the minimum value is 47 years, the median is 63 year and the maximum value is 80 year. Also, there is an outlier at 83.
In the (level 3) the minimum value is 43 years, the median is 63 year and the maximum value is 73 year.
In the (level 4) the minimum value is 23 years, the median is 58 year and the maximum value is 78 year.
In the (level 5) the minimum value is 60 years, the median is 61 year and the maximum value is 63 year **.**

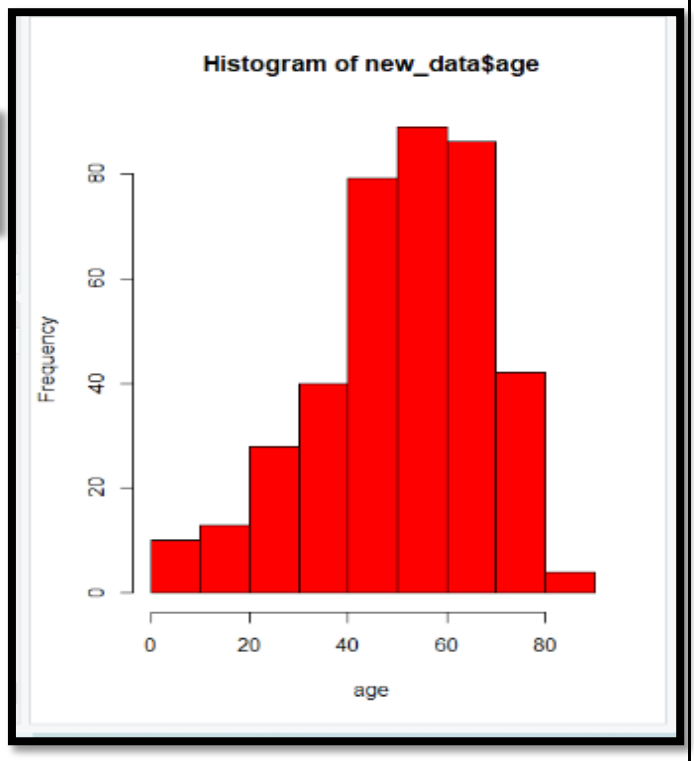
```
# 5-visualize the data in three different ways.

# boxplot
boxplot(age ~ su, data = new_data, xlab = "sugar",
        ylab = "age", main = "age vs sugar")
```



- **Histogram**

This histogram is right skewed, and most of the values are clustered on the left side of the histogram. The highest frequency is from 50 to 60 and the lowest frequency is from 81 to 90.

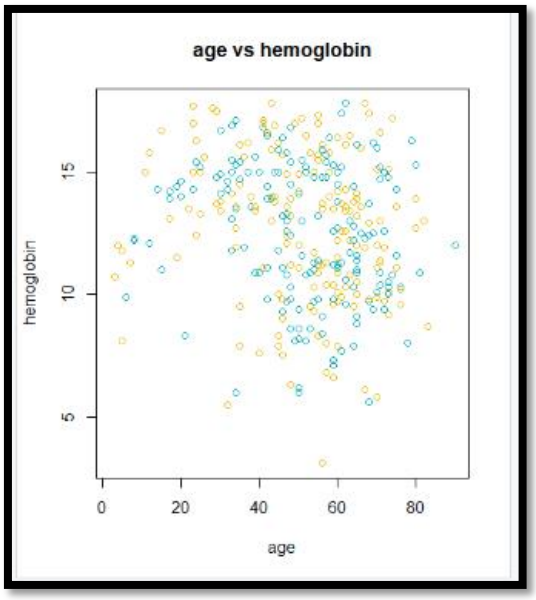
```
#histogram
hist(new_data$age, xlab="age", col= "red")
```



- **Scatter plot**

Most people with kidney disease will develop anemia. Anemia can happen early during kidney disease, so we plot the graph to see if there a correlation or not. The graph shows there is a weak correlation between age and hemoglobin.

```
#Scatterplot
my_cols <- c("#00AFBB", "#E7B800")
plot(x = new_data$age, y = new_data$hemo,
     xlab = "age",
     ylab = "hemoglobin",
     main = "age vs hemoglobin",col = my_cols
)
```



Task 2: Hypothesis Testing

1. Write a hypothesis that you would like to verify.

H0: **There was no difference** in the mean of sugar levels between the person with chronic kidney disease and a person without.

H1: **There is difference** in the mean of sugar levels between the person with chronic kidney disease and a person without.

2. Test your hypothesis using an appropriate test

Anova test is the best statistical test method to infer that there is or isn't a statistical significance because I have more than 2 sample

3. What conclusion can you infer from the sample?

0.00000000125 < 0.01 (0.01 because this is medical situation, so we want less error)
I will reject the null hypothesis and there is significant difference between the mean

```
#  TASK 2: Hypothesis Testing


hypo_testing <- aov(su~classification, data=updated_data)
summary(hypo_testing)
```

```
> hypo_testing <- aov(su~classification, data=updated_data)
> summary(hypo_testing)
                Df Sum Sq Mean Sq F value   Pr(>F)
classification   1   38.8   38.25    38.7 1.25e-09 ***
Residuals      398  393.3    0.99
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Task 3: Model Building & Evaluation

1. Use at least 6 variables to predict whether the person will develop a CDK or not, using:

We choose these variables

	age	bp	bu	sc	wc	rc	cad	classification
1	48	80	36.0	1.20	74	36	0	1
2	7	50	18.0	0.80	58	1	0	1
3	62	80	53.0	1.80	72	1	0	1
4	48	70	56.0	3.80	64	21	0	1
5	51	80	26.0	1.40	70	29	0	1
6	60	90	25.0	1.10	74	27	0	1
7	68	70	54.0	24.00	1	1	0	1
8	24	80	31.0	1.10	66	33	0	1
9	52	100	60.0	1.90	90	23	0	1
10	53	90	107.0	7.20	20	19	0	1
11	50	60	55.0	4.00	1	1	0	1
12	63	70	60.0	2.70	45	20	0	1
13	68	70	72.0	2.10	21	16	1	1
14	68	70	86.0	4.60	1	1	1	1
15	68	80	90.0	4.10	12	7	1	1
16	40	80	162.0	9.60	41	9	0	1
17	47	70	46.0	2.20	1	1	0	1
18	47	80	87.0	5.20	1	1	0	1
19	60	100	27.0	1.30	15	26	1	1
20	62	60	31.0	1.60	51	19	1	1
21	61	80	148.0	3.90	86	14	1	1
22	60	90	180.0	76.00	59	18	1	1
23	48	80	163.0	7.70	66	16	0	1
24	21	70	42.0	1.30	1	1	0	1
25	42	100	50.0	1.40	79	29	0	1
26	61	60	75.0	1.90	80	19	0	1
27	75	80	45.0	2.40	6	22	0	1
28	69	70	87.0	2.70	90	24	1	1
29	75	70	31.0	1.40	1	1	0	1
30	68	70	28.0	1.40	1	1	1	1
31	55	70	155.0	7.30	1	1	0	1
32	73	90	33.0	1.50	74	22	0	1
33	61	90	39.0	1.50	90	23	0	1
34	60	100	55.0	2.50	1	1	0	1
35	70	70	153.0	5.20	1	1	0	1
36	65	90	39.0	2.00	92	32	0	1
37	76	70	29.0	1.80	1	1	0	1
38	72	80	65.0	3.40	66	6	0	0
39	69	80	103.0	4.10	1	1	0	1
40	82	80	70.0	3.40	92	25	0	1
41	46	90	80.0	2.10	85	24	0	1

A. Regression method

a. What type of regression will you use? why?

Logistic regression because we use classification to predict whether the person will develop a CKD or not.

b. Does the data need any preparation for this algorithm? What did you do? Why?

Yes

We fill the numeric missing values by median, and convert the categorical variable to 0 and 1, and drop the id column because it is not an important variable

We did the preparation step to the data, to be suitable to fit the model

```
57
58  # fill the NA of numerical varible with median
59  new_data$bp[is.na(new_data$bp)]<-median(new_data$bp,na.rm=TRUE)
60  new_data
61
62  new_data$sg[is.na(new_data$sg)]<-median(new_data$sg,na.rm=TRUE)
63  new_data
64
65  new_data$su[is.na(new_data$su)]<-median(new_data$su,na.rm=TRUE)
66  new_data
67
68  new_data$bgr[is.na(new_data$bgr)]<-median(new_data$bgr,na.rm=TRUE)
69  new_data
70
```

```
#convert categorical varible to binary ( 0 and 1 )

new_data$rbc <- ifelse(new_data$rbc == "normal",1,0)
new_data

new_data$pcc <- ifelse(new_data$pcc == "present",1,0)
new_data

new_data$ba <- ifelse(new_data$ba == "present",1,0)
new_data

new_data$htn <- ifelse(new_data$htn == "yes",1,0)
new_data

new_data$dm <- ifelse(new_data$dm == "yes",1,0)
new_data
```

```
> updated_data
    age  bp    sg su rbc pcc ba bgr  bu   sc   sod pot  hemo pcv    wc  rc htn dm cad appet pe ane classification
1    48  80 1.020  0   0   0  0 121  36  1.2 138.0 4.4 15.40  44  7800 5.2   1  1   0     1  0   0              1
2     7  50 1.020  0   0   0  0 121  18  0.8 138.0 4.4 11.30  38  6000       0  0   0     1  0   0              1
3    62  80 1.010  3   1   0  0 423  53  1.8 138.0 4.4  9.60  31  7500       0  1   0     0  0   1              1
4    48  70 1.005  0   1   1  0 117  56  3.8 111.0 2.5 11.20  32  6700 3.9   1  0   0     0  1   1              1
5    51  80 1.010  0   1   0  0 106  26  1.4 138.0 4.4 11.60  35  7300 4.6   0  0   0     1  0   0              1
6    60  90 1.015  0   0   0  0  74  25  1.1 142.0 3.2 12.20  39  7800 4.4   1  1   0     1  1   0              1
7    68  70 1.010  0   0   0  0 100  54 24.0 104.0 4.0 12.40  36             0  0   0     1  0   0              1
8    24  80 1.015  4   1   0  0 410  31  1.1 138.0 4.4 12.40  44  6900   5   0  1   0     1  1   0              1
9    52 100 1.015  0   1   1  0 138  60  1.9 138.0 4.4 10.80  33  9600 4.0   1  1   0     1  0   1              1
10   53  90 1.020  0   0   1  0  70 107  7.2 114.0 3.7  9.50  29 12100 3.7   1  1   0     0  0   1              1
11   50  60 1.010  4   0   1  0 490  55  4.0 138.0 4.4  9.40  28             1  1   0     1  0   1              1
12   63  70 1.010  0   0   1  0 380  60  2.7 131.0 4.2 10.80  32  4500 3.8   1  1   0     0  1   0              1
13   68  70 1.015  1   0   1  0 208  72  2.1 138.0 5.8  9.70  28 12200 3.4   1  1   1     0  1   0              1
14   68  70 1.020  0   0   0  0  98  86  4.6 135.0 3.4  9.80                 1  1   1     0  1   0              1
15   68  80 1.010  2   1   1  1 157  90  4.1 130.0 6.4  5.60  16 11000 2.6   1  1   1     0  1   0              1
16   40  80 1.015  0   0   0  0  76 162  9.6 141.0 4.9  7.60  24  3800 2.8   1  0   0     1  0   1              1
17   47  70 1.015  0   0   0  0  99  46  2.2 138.0 4.1 12.60                 0  0   0     1  0   0              1
18   47  80 1.020  0   0   0  0 114  87  5.2 139.0 3.7 12.10                 1  0   0     0  0   0              1
19   60 100 1.025  3   0   0  0 263  27  1.3 135.0 4.3 12.70  37 11400 4.3   1  1   1     1  0   0              1
20   62  60 1.015  0   0   1  0 100  31  1.6 138.0 4.4 10.30  30  5300 3.7   1  0   1     1  0   0              1
21   61  80 1.015  0   0   0  0 173 148  3.9 135.0 5.2  7.70  24  9200 3.2   1  1   1     0  1   1              1
22   60  90 1.020  0   0   0  0 121 180 76.0   4.5 4.4 10.90  32  6200 3.6   1  1   1     1  0   0              1
23   48  80 1.025  0   1   0  0  95 163  7.7 136.0 3.8  9.80  32  6900 3.4   1  0   0     1  0   1              1
24   21  70 1.010  0   0   0  0 121  42  1.3 138.0 4.4 12.65                 0  0   0     0  0   1              1
25   42 100 1.015  0   1   0  1 121  50  1.4 129.0 4.0 11.10  39  8300 4.6   1  0   0     0  0   0              1
26   61  60 1.025  0   0   0  0 108  75  1.9 141.0 5.2  9.90  29  8400 3.7   1  1   0     1  0   1              1
27   75  80 1.015  0   0   0  0 156  45  2.4 140.0 3.4 11.60  35 10300   4   1  1   0     0  0   0              1
28   69  70 1.010  4   1   0  0 264  87  2.7 130.0 4.0 12.50  37  9600 4.1   1  1   1     1  1   0              1
29   75  70 1.020  3   0   0  0 123  31  1.4 138.0 4.4 12.65                 0  1   0     1  0   0              1
30   68  70 1.005  0   0   1  0 121  28  1.4 138.0 4.4 12.90  38             0  0   1     1  0   0              1
31   55  70 1.020  0   0   0  0  93 155  7.3 132.0 4.9 12.65                 1  0   0     1  0   0              1
32   73  90 1.015  0   0   1  0 107  33  1.5 141.0 4.6 10.10  30  7800   4   0  0   0     0  0   0              1
33   61  90 1.010  1   0   0  0 159  39  1.5 133.0 4.9 11.30  34  9600 4.0   1  1   0     0  0   0              1
34   60 100 1.020  0   0   0  0 140  55  2.5 138.0 4.4 10.10  29             1  0   0     0  0   0              1
35   70  70 1.010  0   1   1  1 171 153  5.2 138.0 4.4 12.65                 0  1   0     0  0   0              1
36   65  90 1.020  1   0   0  0 270  39  2.0 138.0 4.4 12.00  36  9800 4.9   1  1   0     0  0   1              1
37   76  70 1.015  0   1   0  0  92  29  1.8 133.0 3.9 10.30  32             1  0   0     1  0   0              1
38   72  80 1.020  0   0   0  0 137  65  3.4 141.0 4.7  9.70  28  6900 2.5   1  1   0     0  0   1              0
39   69  80 1.020  0   0   0  0 121 103  4.1 132.0 5.9 12.50                 1  0   0     1  0   0              1
40   82  80 1.010  2   1   0  0 140  70  3.4 136.0 4.2 13.00  40  9800 4.2   1  1   0     1  0   0              1
41   46  90 1.010  0   1   0  0  99  80  2.1 138.0 4.4 11.10  32  9100 4.1   1  0   0     1  0   0              1
42   45  70 1.010  0   0   0  0 121  20  0.7 138.0 4.4 12.65                 0  0   0     1  1   0              1
43   47 100 1.010  0   0   0  0 204  29  1.0 139.0 4.2  9.70  33  9200 4.5   1  0   0     1  0   1              1
 [ reached 'max' / getOption("max.print") -- omitted 357 rows ]
>
```

```
#   LOGISTIC REGRESSION :


L_data<-updated_data[,c(-3,-4,-5,-6,-7,-8,-11,-12,-13,-14,-17,-18,-20,-21,-22)]
L_data


#split the data into 2 sets, first one takes 70% of the data which is a training set
# the second takes 30% which is a testing set

set.seed(2)
random <- sample(2, nrow(L_data), replace = T, prob = c(0.7, 0.3))
d_train <- L_data[random == 1,]
d_test <- L_data[random == 2,]
d_train
d_test

#fit the Logistic Regression method

logistic <- glm(classification~.,data =d_train, family= binomial)
summary(logistic)
```

In this model, we want to see if the age, blood pressure, serum creatinine, blood urea, white blood cell count, red blood cell count and coronary artery disease has an impact on the CKF (binary classification).

First, we delete the data that we do not use in Logistic regression. *Line 162*
Then, we split the data into training set and testing set as line 169 to 172.

```
> logistic <- glm(classification~.,data =d_train, family= binomial)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(logistic)

Call:
glm(formula = classification ~ ., family = binomial, data = d_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.0016  -0.3045   0.0000   0.1915   2.3362

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.975e+00  1.953e+00  -2.547  0.01085 *
age         -8.451e-03  1.470e-02  -0.575  0.56534
bp           5.514e-02  2.412e-02   2.286  0.02223 *
bu           2.498e-03  1.967e-02   0.127  0.89895
sc           2.845e+00  7.504e-01   3.792  0.00015 ***
wc          -1.035e-03  8.514e-03  -0.122  0.90328
rc          -9.832e-02  1.879e-02  -5.233 1.66e-07 ***
cad          1.619e+01  1.526e+03   0.011  0.99153
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 376.06  on 279  degrees of freedom
Residual deviance: 121.01  on 272  degrees of freedom
AIC: 137.01

Number of Fisher Scoring iterations: 18
```

c. Discuss the result.

The output displays the Deviance Residuals which contains Minimum, 1st quartile, Median, 3rd quartile, and Maximum

The first coefficient is the y-axis intercept when all inputs equal zero so we will get the exponent of this number as $e^{-4.975}$ which equals $6.9*10^{-3}$, for the same bp ,bu , sc ,wc ,rc and cad the odds-ratio of injury CKD is $e^{-0.008451}$ for every unit increase in the age, for the same age ,bu , sc ,wc ,rc and cad the odds-ratio of injury CKD is $e^{0.05514}$ for every unit increase in the bp, for the same age ,bp , sc ,wc ,rc and cad the odds-ratio of injury CKD is $e^{0.002498}$ for unit increase in the bu, for the same age ,bp , bu,wc ,rc and cad the odds-ratio of injury CKD is $e^{2.845}$ for unit increase in the sc,
for the same age, bp , bu, sc, rc and cad the odds-ratio of injury CKD is $e^{-0.001035}$ for unit increase in the wc, for the same age ,bp , bu, sc, wc and cad the odds-ratio of injury CKD is $e^{-0.09832}$ for unit increase in the rc,for the same age ,bp , bu, sc, wc and rc the odds-ratio of injury CKD is $e^{0.1619}$ for unit increase in the cad .

i. Which variables are important/non-important?

- The important variables: First, we can see that blood pressure(bp), serum creatinine (sc)and red blood cell count (rc) are statistically significant predictors ($p < 0.05$). Altitude has the lower p-value suggesting this predictor has a strong association with the CKD occurrence.
- The non-important age, blood urea, white blood cell count and coronary artery disease are statistically significant predictors ($p > 0.05$)

ii. Which variables have a positive/negative impact on the outcome?

- The variables have a positive impact on the outcome is: blood pressure(bp), serum creatinine (sc), blood urea (bu) and coronary artery disease(cad).
- The variables have a negative impact on the outcome is: age, white blood cell count (wc) and red blood cell count (rc).

iii. Pick two variables and write a complete interpretation/explanation for its importance, and its effects on the outcome.

- Age: it has statistically significant predictors (p >0.05) that mean it is a non-important variable and it has a negative impact on the outcome.
- Blood pressure(bp): it has statistically significant predictors (p <0.05) that means it is an important variable and it has a positive impact on the outcome.

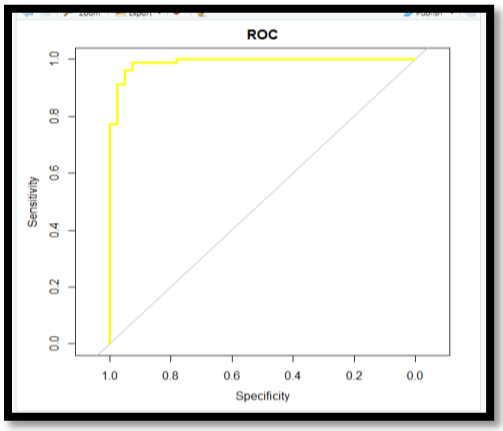
```
180
181
182 #prediction
183 pred <- predict(logistic, d_test, type = "response")
184 pred
185
```

```
> #prediction
> pred <- predict(logistic, d_test, type = "response")
> pred
          2           5           6           8          13          16          17          23          29          33          34          37
0.471420108 0.532430930 0.485409238 0.294803697 0.999999995 1.000000000 0.991531020 0.999999997 0.901356360 0.815028646 0.999241427 0.965680167
         41          45          46          47          50          54          56          57          58          60          61          62
0.965321209 0.999999959 0.999999874 0.257455690 0.995431378 0.228654668 0.945048830 1.000000000 1.000000000 0.999999990 0.999814454 1.000000000
         65          76          78          86          87          89          91         103         104         107         112         115
0.935581242 0.790199105 0.996334895 0.929344239 0.987408676 0.982232286 1.000000000 0.983327266 1.000000000 0.999999961 1.000000000 0.966958059
        116         117         119         123         124         126         132         134         138         146         153         155
0.280395135 0.445976604 0.857824969 1.000000000 0.895492156 0.998473150 0.351392094 0.999970180 0.999914086 1.000000000 0.686090067 1.000000000
        157         161         162         164         168         171         174         176         181         183         184         186
0.999864986 0.999999996 0.416536222 0.967133647 0.744201436 0.626858565 0.999999996 0.887148243 0.991976125 0.760400429 0.999999969 0.743928377
        188         193         201         204         206         211         213         215         222         226         227         229
0.791879162 0.960944495 0.997108634 0.999999994 0.986809890 1.000000000 1.000000000 0.141078803 0.958496144 0.990318081 0.999999994 0.996016442
        230         233         235         241         244         245         246         251         255         261         265         269
1.000000000 0.963365071 0.596228907 0.506120117 0.878221462 0.990312767 1.000000000 0.444557804 0.033970121 0.167150815 0.158457946 0.215693757
        271         272         273         275         277         282         286         289         293         295         301         304
0.277350899 0.102278108 0.186105146 0.772993750 0.128668602 0.129737161 0.019563684 0.180529230 0.066741764 0.012188410 0.014648462 0.053363824
```

We use predict() function to predict the probability That 7 values that we chose will impact the in injury CKD , for example in the first row in the dataset when the age is 48 and bp is 80 and bu 36 and sc 1.2 and wc 7800 and rc 5.2 the predicted value is 0.471420108.

d. Plot ROC curve. Report AUC.

```
201
202 library(pROC)
203 auc(d_test$classification, pred)
204
205 plot(roc(d_test$classification, pred, direction="<"),
206      col="yellow", lwd=3, main="ROC")
207
208
```



The way the curve in the graph points towards the true positive rate (sensitivity) Indicates the number of predictions that were correctly predicted by the model, there is no data going towards the false positive rate(1- sensitivity) this proportion of the graph means the number of observations that were predicted incorrectly.

```
Area under the curve: 0.9895
> plot(roc(d_test$classification, pred, direction="<"),
+      col="yellow", lwd=3, main="ROC")
```

The area under the curve is 98% which means 98% of the predicted data were predicted correctly and it indicates that the model's predictions are accurate.

e. Evaluate the developed model using at least three different metrics.

```
185
186 #Testing
187 library(Metrics)
188
189 #Mean Squared Error (MSE)
190 mean_squared_error <- mse(L_data$classification, pred)
191 mean_squared_error
192
193 #Mean Absolute Error (MAE)
194 mean_abs_error <- mae(L_data$classification, pred)
195 mean_abs_error
196
197 #Mean Absolute Deviation
198 mean_abs_deviation <- mean_abs_error/299
199 mean_abs_deviation
200
```

i. Explain each metric. What is it and how is it calculated?

- Mean Squared Error (MSE)
  It calculates the average squared difference between the predicted values and the actual value.
- Mean Absolute Error (MAE)
  is the average magnitude of the errors in the prediction.
- Mean Absolute Deviation
  is the average distance between each data point and the mean.

ii. What value did you get?

```
> mean_squared_error
[1] 0.4152926
> #Mean Absolute Error (MAE)
> mean_abs_error <- mae(L_data$classification, pred)
Warning message:
In actual - predicted :
  longer object length is not a multiple of shorter object length
> mean_abs_error
[1] 0.4884297
> #Mean Absolute Deviation
> mean_abs_deviation <- mean_abs_error/299
> mean_abs_deviation
[1] 0.001633544
> library(pROC)
> auc(d_test$classification, pred)
```


iii. What does that mean?

- The output of (MSE) came out as 0.415, The lower the value and close to 0 the more perfect the model, so this is an indication that our model is good
- The output of (MAE)came out as 0.49, which is the average magnitude of the errors in the predictions, is good.
- The output of mean absolute deviation came out as 0.0016, which means there is no obvious distance, and this is an indication that our model is good

B. Naïve Bayes

a. Does the data need any preparation for this algorithm? What did you do? Why?

We clean the data before the model building step, as we mention in the regression step

b. Evaluate the developed model using at least three different metrics.

first in the *239 line* we drop the column that does not serve us
and in the *244 line* we convert the classification column to factor
and from *253 line* to *258 line* we split the data by 70% for training and 30% for testing
in the *262 line* we use naïve bayes library to train the data
and in the *266 line* we predict data
and finally, in the *270 line* we use confusion matrix to print the evaluation matrix

i. What value did you get?
ii. What does that mean?

Accuracy = 70%, which is good, and it means the number of correctly predicted data points out of all data points
Precision = 53,25%, this percentage means 53% of the data were positively predicted
Recall = 100%, which means the percentage of total true positive data were classified as true positive

```
> #print metrics
> confusionMatrix(pred_nb, data_test$classification)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41 36
         1  0 43

               Accuracy : 0.7
                 95% CI : (0.6096, 0.7802)
    No Information Rate : 0.6583
    P-Value [Acc > NIR] : 0.1939

                  Kappa : 0.4494

 Mcnemar's Test P-Value : 5.433e-09

            Sensitivity : 1.0000
            Specificity : 0.5443
         Pos Pred Value : 0.5325
         Neg Pred Value : 1.0000
             Prevalence : 0.3417
         Detection Rate : 0.3417
   Detection Prevalence : 0.6417
      Balanced Accuracy : 0.7722

       'Positive' Class : 0

>
```

```
236
237
238
239 NB_data <- updated_data[, c(-3,-4-5,-6,-7,-8,-11,-12,-13,-14,-17,-18,-20,-21,-22)]
240 NB_data
241
242 str(NB_data)
243
244 NB_data$classification <- as.factor(NB_data$classification)
245 str(NB_data)
246
247
248 library(e1071)
249 library(caret)
250
251 #split the data into 2 sets, first one takes 70% of the data which is a training set
252 # the second takes 30% which is a testing set
253 set.seed(2)
254 random <- sample(2, nrow(NB_data), prob = c(0.7, 0.3), replace = T)
255 data_train <- NB_data[random == 1, ]
256 data_test <- NB_data[random == 2, ]
257 data_train
258 data_test
259
260
261 #fit the naive bayes method
262 data_nb <- naiveBayes(classification ~ . , data  = data_train)
263 data_nb
264
265 #predict
266 pred_nb <- predict(data_nb, data_test)
267 pred_nb
268
269 #print metrics
270 confusionMatrix(pred_nb, data_test$classification)
271
272
273
```

C. Decision Trees

a. Discuss the result (by interpreting the output of the model).

a. Does the data need any preparation for this algorithm? What did you do? Why?

We cleaned the data before the model building step, as we mentioned in the regression step. We also converted the classification event attribute to a factor with two levels (Yes & No) which means when the classification event is equal 1 the kidney disease happened and it's Yes, when the classification event is equal 0 it will be No it didn't happen. Then we split the data into 2 sets, first one takes 70% of the data which is a training set

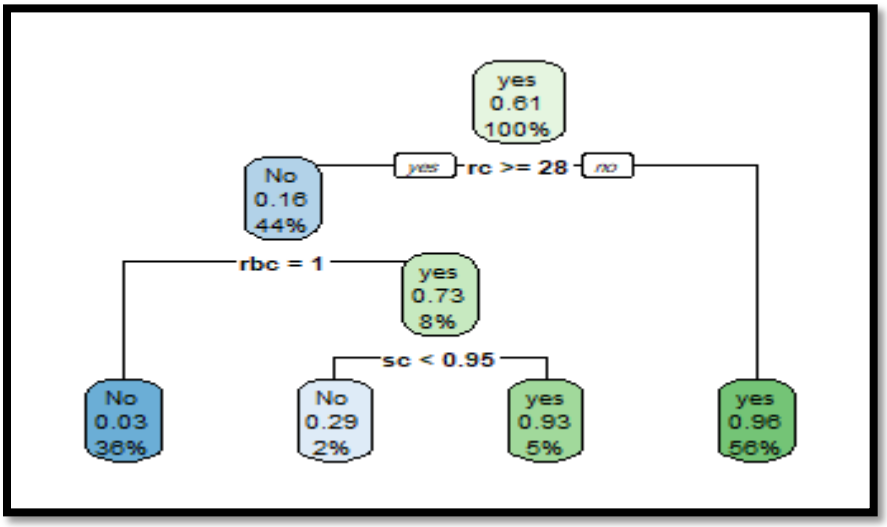
The second takes 30% which is a testing set

```
#  convert the classification ( ckd or not)  to factor (yes & no)
DT_data[DT_data$classification ==1,]$classification = "yes"
DT_data[DT_data$classification ==0,]$classification = "No"

DT_data$classification = as.factor(DT_data$classification)
str(DT_data)
```

```
DT = sample(2, No_Obs, replace = TRUE, prob = c(0.7,0.3))
DT_training_set <- DT_data[DT==1, ]
DT_testing_set <- DT_data[DT ==2, ]
```

b. Draw the tree and write at least two rules.

```
DT_testing_set <- DT_data[DT ==2, ]

#draw the tree using the training set
library(rpart)
library(rpart.plot)

model_DT = rpart(DT_training_set$classification~.,DT_training_set)
print(model_DT)

# visualize the tree
rpart.plot(model_DT)
```



1- If rc (red blood cell count) < 28 then there's a high chance of 56% this person will get kidney disease
2- If rc (red blood cell count) >= 28 and rbc (red blood cells) does not equal 1, then there is a low chance of 36% that this person will get kidney disease

Then we imported the library rpart and rpart.plot in order to draw the tree, and we create the variable model_DT that contains the classification attribute which is our class label and the training set. The output explains how the tree will be branched.

After that we visualized the tree using rpart.plot() function. the root node is the rc (red blood cell count) < 28. the right-side sub-tree is a leaf node of rc (red blood cell count) >= 28. and the left side sub-tree is a decision node of rbc (red blood cells)=1. And the other decision node is when the sc (serum creatinine) < 0.95.

c. What are the variables that best split the data in the first and second level of the tree?

In the first level the best node is rc (red blood cell count)
In the second level the best node is rbc (red blood cells)

d. Evaluate the developed model using at least three different metrics.
   i. What values did you get?
   ii. What does that mean?

```
> #Evaluate the model using confusion matrix
> CM = confusionMatrix(test_result,DT_testing_set$classification)
> CM
Confusion Matrix and Statistics

          Reference
Prediction No yes
       No  46   6
       yes  4  72

               Accuracy : 0.9219
                 95% CI : (0.861, 0.9619)
    No Information Rate : 0.6094
    P-Value [Acc > NIR] : 8.907e-16

                  Kappa : 0.8371

 Mcnemar's Test P-Value : 0.7518

            Sensitivity : 0.9200
            Specificity : 0.9231
         Pos Pred Value : 0.8846
         Neg Pred Value : 0.9474
             Prevalence : 0.3906
         Detection Rate : 0.3594
   Detection Prevalence : 0.4062
      Balanced Accuracy : 0.9215

       'Positive' Class : No

> conf <- table(actual = DT_testing_set$classification, predicted= test_result )
> accuracy= sum(diag(conf))/ sum(conf)
> print(accuracy)
[1] 0.921875
> percision= conf[1,1]/colSums(conf)[1]
> print(percision)
       No
0.8846154
> recall_DT = conf[1,1]/rowSums(conf)[1]
> print(recall_DT)
   No
0.92
> |
```

We calculate the confusion matrix to evaluate the decision tree.

1- The accuracy is equal to 92% which calculates the overall success rate, it's all the actual kidney disease cases that was

predicted as yes (TP) plus all the non-kidney disease event that was predict as No (TN), divided by all the predicted data. Based on the calculated value we can conclude that the accuracy of the model is very good identifying relationship between variables.

2- The precision is equal 88%, which is all the actual kidney disease cases that was predicted as yes (TP) divided by all actual kidney disease cases that was predicted Yes (TP) plus all the non- kidney disease cases that was predicted as Yes (FP).

3- The recall is equal 92%, which is all the actual kidney disease cases that was predicted as yes (TP) divided by all actual kidney disease cases that was predicted Yes (TP) plus all actual kidney disease cases that was predicted as non- kidney disease event (FN).

2. Which method performed the best?

The result shows the logistic regression model is the best with AUC 98%
And for us it was easy to implement it, and we did not face any problems with it.
And now we have the experience to use the knowledge we had from this course to any real-world problem.