Pre-processing and Data Visualization: Chronic Kidney Disease:

Submitted by Rianna Aalto Instructor: Ishita Jaju

This task is made of two parts:

- Pre-processing
- Data Visualization

These are library.package that are being used: library(tidyverse), library(forcats), ggplot2

A brief description:

DATA:

This dataset was suggested by our instructor for our disposal therefore I used it without hesitation. I downloaded the dataset through the given link and saved it as csv file in my computer: https://www.kaggle.com/mahmoudlimam/preprocessed-chronic-kidney-disease-dataset

The dataset contains 26 variables/ features and 400 observations which may predict a patient with chronic kidney disease. Out of the 26 features, there are 14 that are numerical and 10 that are categorical.

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod
1	0	48	80	1.020	1	0		normal	notpresent	notpresent	121	36.0	1.20	
2	1	7	50	1.020	4	0		normal	notpresent	notpresent	NA	18.0	0.80	
3	2	62	80	1.010	2	3	normal	normal	notpresent	notpresent	423	53.0	1.80	
4	3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56.0	3.80	1
5	4	51	80	1.010	2	0	normal	normal	notpresent	notpresent	106	26.0	1.40	
6	5	60	90	1.015	3	0			notpresent	notpresent	74	25.0	1.10	1
7	6	68	70	1.010	0	0		normal	notpresent	notpresent	100	54.0	24.00	1
8	7	24	NA	1.015	2	4	normal	abnormal	notpresent	notpresent	410	31.0	1.10	
9	8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60.0	1.90	
10	9	53	90	1.020	2	0	abnormal	abnormal	present	notpresent	70	107.0	7.20	1
11	10	50	60	1.010	2	4		abnormal	present	notpresent	490	55.0	4.00	
12	11	63	70	1.010	3	0	abnormal	abnormal	present	notpresent	380	60.0	2.70	1
13	12	68	70	1.015	3	1		normal	present	notpresent	208	72.0	2.10	1
14	13	68	70	NA	NA	NA			notpresent	notpresent	98	86.0	4.60	1
15	14	68	80	1.010	3	2	normal	abnormal	present	present	157	90.0	4.10	1

Go forward to the next source location (Ctrl+F10)

Showing 1 to 15 of 400 entries, 26 total columns

• age - age • bp - blood pressure • sg - specific gravity • al - albumin • su - sugar • rbc - red blood cells • pc - pus cell • pcc - pus cell clumps • ba - bacteria • bgr - blood glucose random • bu - blood urea • sc - serum creatinine • sod - sodium • pot - potassium	• hemo - hemoglobin • pcv - packed cell volume • wc - white blood cell count • rc - red blood cell count • htn - hypertension • dm - diabetes mellitus • cad - coronary artery disease • appet - appetite • pe - pedal edema • ane - anemia • class - class

PRE-PROCESSING

I. IMPORTING downloaded dataset from my computer and understanding the content.
Syntax:
> kidney.disease = read.csv('…kidney.disease.csv', stringsAsFactors = FALSE)
> View(kidney.disease)

II. CHECKING the structure of data and coming up with the list of problems

Command: str(kidney.disease)

```
'data.frame':   400 obs. of  26 variables:
 $ id             : chr  "0" "1" "2" "3" ...
 $ 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 ...
 $ al             : num  1 4 2 4 2 3 0 2 3 2 ...
 $ su             : num  0 0 3 0 0 0 0 4 0 0 ...
 $ rbc            : chr  "" "" "normal" "normal" ...
 $ pc             : chr  "normal" "normal" "normal" "abnormal" ...
 $ 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            : int  44 38 31 32 35 39 36 44 33 29 ...
 $ wc             : int  7800 6000 7500 6700 7300 7800 NA 6900 9600 12100 ...
 $ rc             : num  5.2 NA NA 3.9 4.6 4.4 NA 5 4 3.7 ...
 $ 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" ...
```

By doing some exploration, I noticed that there are many values missing, missed placed and/ wrongly distributed and classified.

- rbc/pc - A large value of the these features are missing or not available
- htn, dm, cad, pe, ane – Values are classified as “character/string” however can be converted to TRUE/FALSE or 1/0
- sod, pot etc – Have values that are missing or contains NA
- Id - Some of the values are missed placed or not distributed/converted correctly in their proper column , misspelled and corrupted values
- Rbc, pc, pcc, ba – have undefined values or missing or corrupted values
- htn, dm, cad, appet, pe, ane, – they have missing values
- classification – needs to be abbreviated
- id – has a variable type, character, should be replaced with numeric

By running sum(as.na), we can see how many data has NA . There are 932 NA in the dataset.

```
> sum(is.na(kidney.disease))
[1] 932
```

I am checking further if all the data provided in each row are correct.

```
levels(as.factor(`kidney.disease.(1)`$id))
```

```
 [38] "131"

 [39] "132"

 [40] "133,70.0,100.0,1.015,4.0,0.0,normal,normal,notpresent,notpresent,118.0,125.0,5.
3,136.0,4.9,12.0,37,\t8400,8.0,yes,no,no,good,no,no,ckd"
 [41] "134"

 [42] "135"

 [43] "136"

 [44] "137"

 [45] "138,73.0,,1.01,1.0,0.0,,,notpresent,notpresent,95.0,51.0,1.6,142.0,3.5,,,,,n
o,\tno,no,good,no,no,ckd"
 [46] "139"

 [47] "14"
```

```
 [72] "162,59.0,70.0,,,,,,notpresent,notpresent,204.0,34.0,1.5,124.0,4.1,9.8,3
7,6000,\t?,no,yes,no,good,no,no,ckd"
 [73] "163"
```

```
 [97] "185,4.0,,1.02,1.0,0.0,,normal,notpresent,notpresent,99.0,23.0,0.6,138.0,
4.4,12.0,34,\t?,,no,no,no,good,no,no,ckd"
 [98] "186"

 [99] "187"

[100] "188,8.0,,,,,,,notpresent,notpresent,80.0,66.0,2.5,142.0,3.6,12.2,38,,,n
o,\tno,no,good,no,no,ckd"
```

```
[107] "194,80.0,70.0,1.01,2.0,,,abnormal,notpresent,notpresent,,49.0,1.2,,,,,,
yes,\tyes,no,good,no,no,ckd"
```

```
[130] "214,68.0,80.0,1.015,0.0,0.0,,abnormal,notpresent,notpresent,171.0,30.0,
1.0,,,13.7,\t43,4900,5.2,no,yes,no,good,no,no,ckd"
```

```
[302] "37,72.0,80.0,,,,,,notpresent,notpresent,137.0,65.0,3.4,141.0,4.7,9.7,28,
6900,2.5,yes,yes,no,poor,no,yes,ckd\t"
```

```
[336] "40,46.0,90.0,1.01,2.0,0.0,normal,abnormal,notpresent,notpresent,99.0,80.0,2.1,,,11.
1,32,9100,4.1,yes,no,\tno,good,no,no,ckd"
```

```
[362] "64,55.0,80.0,1.01,0.0,0.0,,normal,notpresent,notpresent,146.0,,,,,9.8,,,,no,no,\tn
o,good,no,no,ckd"
[363] "65,44.0,90.0,1.01,1.0,0.0,,normal,notpresent,notpresent,,20.0,1.1,,,15.0,48,,,no,\t
no,no,good,no,no,ckd"
[364] "66,67.0,70.0,1.02,2.0,0.0,abnormal,normal,notpresent,notpresent,150.0,55.0,1.6,131.
0,4.8,,\t?,,,yes,yes,no,good,yes,no,ckd"
```

```
[375] "76,48.0,80.0,1.005,4.0,0.0,abnormal,abnormal,notpresent,present,133.0,139.0,8.5,13
2.0,5.5,10.3,36,\t6200,4,no,yes,no,good,yes,no,ckd"
```

```
[388] "88,58.0,110.0,1.01,4.0,0.0,,normal,notpresent,notpresent,251.0,52.0,2.2,,,,,13200,
4.7,yes,\tyes,no,good,no,no,ckd"
```

```
> levels(as.factor(kidney.disease$rbc))
[1] ""         "abnormal" "normal"
> levels(as.factor(kidney.disease$pc))
[1] ""         "abnormal" "normal"
>
> > levels(as.factor(kidney.disease$pcc))
Error: unexpected '>' in ">"
>  levels(as.factor(kidney.disease$pcc))
[1] ""           "notpresent" "present"
>
> levels(as.factor(kidney.disease$ba))
[1] ""           "notpresent" "present"
```

```
> levels(as.factor(kidney.disease$age))
 [1] "2"  "3"  "5"  "6"  "7"  "8"  "11" "12" "14" "15" "17" "19" "20" "21" "22" "23"
[17] "24" "25" "26" "27" "28" "29" "30" "32" "33" "34" "35" "36" "37" "38" "39" "40"
[33] "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54" "55" "56"
[49] "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72"
[65] "73" "74" "75" "76" "78" "79" "80" "81" "82" "83" "90"
> levels(as.factor(kidney.disease$bp))
 [1] "50"  "60"  "70"  "80"  "90"  "100" "110" "120" "140" "180"
> levels(as.factor(kidney.disease$sg))
[1] "1.005" "1.01"  "1.015" "1.02"  "1.025"
> levels(as.factor(kidney.disease$al))
[1] "0" "1" "2" "3" "4" "5"
> levels(as.factor(kidney.disease$su))
[1] "0" "1" "2" "3" "4" "5"
> levels(as.factor(kidney.disease$rbc))
[1] ""         "abnormal" "normal"
> levels(as.factor(kidney.disease$pc))
[1] ""         "abnormal" "normal"
>
> > levels(as.factor(kidney.disease$pcc))
Error: unexpected '>' in ">"
>  levels(as.factor(kidney.disease$pcc))
[1] ""           "notpresent" "present"
>
> levels(as.factor(kidney.disease$ba))
[1] ""           "notpresent" "present"
```

```
> levels(as.factor(kidney.disease$bgr))
  [1] "22"  "70"  "74"  "75"  "76"  "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"
 [14] "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"
 [27] "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108" "109" "110" "111"
 [40] "112" "113" "114" "115" "116" "117" "118" "119" "120" "121" "122" "123" "124"
 [53] "125" "127" "128" "129" "130" "131" "132" "133" "134" "137" "138" "139" "140"
 [66] "141" "143" "144" "148" "150" "153" "156" "157" "158" "159" "160" "162" "163"
 [79] "165" "169" "171" "172" "173" "176" "182" "184" "192" "201" "203" "204" "207"
 [92] "208" "210" "213" "214" "215" "219" "220" "224" "226" "230" "233" "234" "238"
[105] "239" "241" "242" "246" "248" "250" "252" "253" "255" "256" "261" "263" "264"
[118] "268" "269" "270" "273" "280" "288" "294" "295" "297" "298" "303" "307" "308"
[131] "309" "323" "341" "352" "360" "380" "410" "415" "423" "424" "425" "447" "463"
[144] "490"
> levels(as.factor(kidney.disease$bu))
  [1] "1.5"  "10"   "15"   "16"   "17"   "18"   "19"   "20"   "21"   "22"   "23"
 [12] "24"   "25"   "26"   "27"   "28"   "29"   "30"   "31"   "32"   "33"   "34"
 [23] "35"   "36"   "37"   "38"   "39"   "40"   "41"   "42"   "44"   "45"   "46"
 [34] "47"   "48"   "49"   "50"   "50.1" "51"   "52"   "53"   "54"   "55"   "56"
 [45] "57"   "58"   "60"   "61"   "64"   "66"   "67"   "68"   "70"   "71"   "72"
 [56] "73"   "74"   "75"   "76"   "77"   "79"   "80"   "82"   "85"   "86"   "87"
 [67] "88"   "89"   "90"   "92"   "93"   "94"   "95"   "96"   "98"   "98.6" "103"
 [78] "106"  "107"  "111"  "113"  "114"  "115"  "118"  "125"  "132"  "133"  "137"
 [89] "142"  "145"  "146"  "148"  "150"  "153"  "155"  "158"  "162"  "163"  "164"
[100] "165"  "166"  "176"  "180"  "186"  "191"  "202"  "208"  "215"  "217"  "219"
[111] "223"  "235"  "241"  "309"  "322"  "391"
```

```
> levels(as.factor(kidney.disease$sc))
 [1] "0.4"  "0.5"  "0.6"  "0.7"  "0.8"  "0.9"  "1"    "1.1"  "1.2"  "1.3"  "1.4"
[12] "1.5"  "1.6"  "1.7"  "1.8"  "1.9"  "2"    "2.1"  "2.2"  "2.3"  "2.4"  "2.5"
[23] "2.6"  "2.7"  "2.8"  "2.9"  "3"    "3.2"  "3.25" "3.3"  "3.4"  "3.6"  "3.8"
[34] "3.9"  "4"    "4.1"  "4.3"  "4.4"  "4.6"  "5.2"  "5.3"  "5.6"  "5.9"  "6"
[45] "6.1"  "6.3"  "6.4"  "6.5"  "6.7"  "6.8"  "7.1"  "7.2"  "7.3"  "7.5"  "7.7"
[56] "9.2"  "9.3"  "9.6"  "9.7"  "10.2" "10.8" "11.5" "11.8" "11.9" "12"   "12.2"
[67] "12.8" "13"   "13.3" "13.4" "13.5" "13.8" "14.2" "15"   "15.2" "16.4" "16.9"
[78] "18"   "18.1" "24"   "32"   "48.1" "76"
> levels(as.factor(kidney.disease$sod))
 [1] "4.5" "104" "111" "113" "114" "115" "120" "122" "124" "125" "126" "127" "128"
[14] "129" "130" "131" "132" "133" "134" "135" "136" "137" "138" "139" "140" "141"
[27] "142" "143" "144" "145" "146" "147" "150" "163"
> levels(as.factor(kidney.disease$pot))
 [1] "2.5" "2.7" "2.8" "2.9" "3"   "3.2" "3.3" "3.4" "3.5" "3.6" "3.7" "3.8" "3.9"
[14] "4"   "4.1" "4.2" "4.3" "4.4" "4.5" "4.6" "4.7" "4.8" "4.9" "5"   "5.1" "5.2"
[27] "5.3" "5.4" "5.5" "5.6" "5.7" "5.8" "5.9" "6.3" "6.4" "6.5" "6.6" "7.6" "39"
[40] "47"
> levels(as.factor(kidney.disease$hemo))
  [1] "3.1"  "4.8"  "5.5"  "5.6"  "5.8"  "6"    "6.1"  "6.2"  "6.3"  "6.6"  "6.8"
 [12] "7.1"  "7.3"  "7.5"  "7.6"  "7.7"  "7.9"  "8"    "8.1"  "8.2"  "8.3"  "8.4"
 [23] "8.5"  "8.6"  "8.7"  "8.8"  "9"    "9.1"  "9.2"  "9.3"  "9.4"  "9.5"  "9.6"
 [34] "9.7"  "9.8"  "9.9"  "10"   "10.1" "10.2" "10.3" "10.4" "10.5" "10.6" "10.7"
 [45] "10.8" "10.9" "11"   "11.1" "11.2" "11.3" "11.4" "11.5" "11.6" "11.7" "11.8"
 [56] "11.9" "12"   "12.1" "12.2" "12.3" "12.4" "12.5" "12.6" "12.7" "12.8" "12.9"
 [67] "13"   "13.1" "13.2" "13.3" "13.4" "13.5" "13.6" "13.7" "13.8" "13.9" "14"
 [78] "14.1" "14.2" "14.3" "14.4" "14.5" "14.6" "14.7" "14.8" "14.9" "15"   "15.1"
 [89] "15.2" "15.3" "15.4" "15.5" "15.6" "15.7" "15.8" "15.9" "16"   "16.1" "16.2"
[100] "16.3" "16.4" "16.5" "16.6" "16.7" "16.8" "16.9" "17"   "17.1" "17.2" "17.3"
[111] "17.4" "17.5" "17.6" "17.7" "17.8"
```

```
> levels(as.factor(kidney.disease$pcv))
 [1] "9"  "14" "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28"
[17] "29" "30" "31" "32" "33" "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44"
[33] "45" "46" "47" "48" "49" "50" "51" "52" "53" "54"
> levels(as.factor(kidney.disease$wc))
 [1] "2200"  "2600"  "3800"  "4100"  "4200"  "4300"  "4500"  "4700"  "5000"  "5100"
[11] "5200"  "5300"  "5400"  "5500"  "5600"  "5700"  "5800"  "5900"  "6000"  "6200"
[21] "6300"  "6400"  "6500"  "6600"  "6700"  "6800"  "6900"  "7000"  "7100"  "7200"
[31] "7300"  "7400"  "7500"  "7700"  "7800"  "7900"  "8000"  "8100"  "8200"  "8300"
[41] "8400"  "8500"  "8600"  "8800"  "9000"  "9100"  "9200"  "9300"  "9400"  "9500"
[51] "9600"  "9700"  "9800"  "9900"  "10200" "10300" "10400" "10500" "10700" "10800"
[61] "10900" "11000" "11200" "11300" "11400" "11500" "11800" "11900" "12000" "12100"
[71] "12200" "12300" "12400" "12500" "12700" "12800" "13200" "13600" "14600" "14900"
[81] "15200" "15700" "16300" "16700" "18900" "19100" "21600" "26400"
> levels(as.factor(kidney.disease$rc))
 [1] "2.1" "2.3" "2.4" "2.5" "2.6" "2.7" "2.8" "2.9" "3"   "3.1" "3.2" "3.3" "3.4"
[14] "3.5" "3.6" "3.7" "3.8" "3.9" "4"   "4.1" "4.2" "4.3" "4.4" "4.5" "4.6" "4.7"
[27] "4.8" "4.9" "5"   "5.1" "5.2" "5.3" "5.4" "5.5" "5.6" "5.7" "5.8" "5.9" "6"
[40] "6.1" "6.2" "6.3" "6.4" "6.5"
> levels(as.factor(kidney.disease$htn))
[1] ""    "no"  "yes"
> levels(as.factor(kidney.disease$dm))
[1] ""     " yes" "no"   "yes"
> levels(as.factor(kidney.disease$cad))
[1] ""    "no"  "yes"
> levels(as.factor(kidney.disease$appet))
[1] ""     "good" "poor"
```

```
> levels(as.factor(kidney.disease$pe))
[1] ""    "no"  "yes"
> levels(as.factor(kidney.disease$ane))
[1] ""    "no"  "yes"
> levels(as.factor(kidney.disease$classification))
[1] ""       "ckd"    "notckd"
```

Here, I checked how many NA values in the dataset and see the number of NA in each column.

```
> sum(is.na(kidney.disease)
```

```
> colSums(is.na(kidney.disease))
            id            age             bp             sg             al
             0             24             24             59             58
            su            rbc             pc            pcc             ba
             0              0              0              0              0
           bgr             bu             sc            sod            pot
            57             33             31             96             97
          hemo            pcv             wc             rc            htn
            62             80            114            137              0
            dm            cad          appet             pe            ane
             0              0              0              0              0
classification
             0
```

III FIXING THE CORRUPTED DATA

A. Changing column types to correct ones
Just for my convenience, calling kidney.disease dataset as Dataset.
#call kidney.disease dataset to Dataset

```
> Dataset <- kidney.disease
> View(Dataset)
```

Change the column type for "id" from character to number
> Dataset$id <- as.numeric(Dataset$id)

```
> str(Dataset)
'data.frame':   400 obs. of  26 variables:
 $ id             : num  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 ...
 $ al             : num  1 4 2 4 2 3 0 2 3 2 ...
 $ su             : num  0 0 3 0 0 0 0 4 0 0 ...
 $ rbc            : chr  "" "" "normal" "normal" ...
 $ pc             : chr  "normal" "normal" "normal" "abnormal" ...
 $ pcc            : chr  "notpresent" "notpresent" "notpresent" "present" ...
 $ ba             : chr  "notpresent" "notpresent" "notpresent" "notpresent"
 ...
```

B. I wanted to distribute or move those data in the rows mention below to the right column but didn't get the working command for "cSlit" so I decided to delete / drop a total of 15 row that were corrupted in variable "id".

```
[336] "40,46.0,90.0,1.01,2.0,0.0,normal,abnormal,notpresent,notpresent,99.0,80.0,2.1,,,11.
1,32,9100,4.1,yes,no,\tno,good,no,no,ckd"
```

```
> Sli1 <- kidney.disease[-c(41,65,66,67,77,89,134,139,163,186,189,195,215,23
1),]
> View(Sli1)
> View(kidney.disease)
```

```
> Sli1$id
  [1] "0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"
 [12] "11"  "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"
 [23] "22"  "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"
 [34] "33"  "34"  "35"  "36"  "38"  "39"  "41"  "42"  "43"  "44"  "45"
 [45] "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"
 [56] "57"  "58"  "59"  "60"  "61"  "62"  "63"  "67"  "68"  "69"  "70"
 [67] "71"  "72"  "73"  "74"  "75"  "77"  "78"  "79"  "80"  "81"  "82"
 [78] "83"  "84"  "85"  "86"  "87"  "89"  "90"  "91"  "92"  "93"  "94"
 [89] "95"  "96"  "97"  "98"  "99"  "100" "101" "102" "103" "104" "105"
[100] "106" "107" "108" "109" "110" "111" "112" "113" "114" "115" "116"
[111] "117" "118" "119" "120" "121" "122" "123" "124" "125" "126" "127"
[122] "128" "129" "130" "131" "132" "134" "135" "136" "137" "139" "140"
[133] "141" "142" "143" "144" "145" "146" "147" "148" "149" "150" "151"
[144] "152" "153" "154" "155" "156" "157" "158" "159" "160" "161" "163"
[155] "164" "165" "166" "167" "168" "169" "170" "171" "172" "173" "174"
[166] "175" "176" "177" "178" "179" "180" "181" "182" "183" "184" "186"
[177] "187" "189" "190" "191" "192" "193" "195" "196" "197" "198" "199"
[188] "200" "201" "202" "203" "204" "205" "206" "207" "208" "209" "210"
```

C. Replace the missing data with the average of the feature in which the data is missing. These are the features/variables that has missing data and by this command, it will show total NA values that are available in each feature.

```
colSums(is.na(Sli1))
```

Output:

```
> colSums(is.na(Sli1))
            id            age             bp             sg             al
             0              9              9             44             43
            su            rbc             pc            pcc             ba
            45              0              0              0              0
           bgr             bu             sc            sod            pot
            42             18             16             81             82
          hemo            pcv             wc             rc            htn
            47             65             99            122              0
            dm            cad          appet             pe            ane
             0              0              0              0              0
classification
             0
```

Variable "age", there are 9 missing (NA) data. To replace the NA value with the average, we will run the command:

```
> mean_age <- as.integer(mean(Sli1$age, na.rm = TRUE))
> Sli1$age[is.na(Sli1$age)] = mean_age
```

Output:

```
> Sli1$age
  [1] 48  7 62 48 51 60 68 24 52 53 50 63 68 68 68 40 47 47 60 62
 [21] 61 60 48 21 42 61 75 69 75 68 51 73 61 60 70 65 76 69 82 45
 [41] 47 35 54 54 48 11 73 60 53 54 53 62 63 35 76 76 73 59 67 67
 [61] 15 46 45 65 26 61 46 64 51 56  5 67 70 56 74 45 38 48 59 70
 [81] 56 70 50 63 56 71 73 65 62 60 65 50 56 34 71 17 76 55 65 50
[101] 55 45 54 63 65 51 61 12 47 51 51 55 60 72 54 34 43 65 72 70
[121] 71 52 75 50  5 50 47 48 46 45 41 69 67 72 41 60 57 53 60 69
[141] 65  8 76 39 55 56 50 66 62 71 59 81 62 46 14 60 27 34 65 51
[161] 66 83 62 17 54 60 21 65 42 72 73 45 61 30 54  8  3 64  6 51
[181] 46 32 70 49 57 59 65 90 64 78 51 65 61 60 50 67 19 59 54 40
[201] 55  2 64 63 33 68 36 66 74 71 34 60 64 57 60 59 60 50 51 37
[221] 45 65 80 72 34 65 57 69 62 64 48 48 54 59 56 40 23 45 57 51
[241] 34 60 38 42 35 30 49 55 45 42 50 55 48 51 25 23 30 56 47 19
[261] 52 20 46 48 24 47 55 20 60 33 66 71 39 56 42 54 47 30 50 75
[281] 44 41 53 34 73 45 44 29 55 33 41 52 47 43 51 46 56 80 55 39
[301] 44 35 58 61 30 57 65 70 43 40 58 47 30 28 33 43 59 34 23 24
[321] 60 25 44 62 25 32 63 44 37 64 22 33 43 38 35 65 29 37 39 32
```

Variable "bp", there are 9 missing (NA) data. Doing the same process to the rest of features that have NA values. It would be great to learn and use the impute function to shorten the process, but this time, I'm using the longer process. And doing the same to the rest of these numerical data.

```
> mean_bp <- as.integer(mean(Sli1$bp, na.rm = TRUE))
> mean_sg <- as.integer(mean(Sli1$sg, na.rm = TRUE))
> mean_al <- as.integer(mean(Sli1$al, na.rm = TRUE))
> mean_su <- as.integer(mean(Sli1$su, na.rm = TRUE))
> mean_bgr <- as.integer(mean(Sli1$bgr, na.rm = TRUE))
> mean_bu <- as.integer(mean(Sli1$bu, na.rm = TRUE))
> mean_sc <- as.integer(mean(Sli1$sc, na.rm = TRUE))
> mean_sod <- as.integer(mean(Sli1$sod, na.rm = TRUE))
> mean_pot <- as.integer(mean(Sli1pot, na.rm = TRUE))
> mean_hemo <- as.integer(mean(Sli1$hemo, na.rm = TRUE))
> mean_pcv <- as.integer(mean(Sli1$pcv, na.rm = TRUE))
> mean_wc <- as.integer(mean(Sli1$wc, na.rm = TRUE))
> mean_rc <- as.integer(mean(Sli1$rc, na.rm = TRUE))
> Sli1$bp[is.na(Sli1$bp)] = mean_bp
> Sli1$sg[is.na(Sli1$sg)] = mean_sg
> Sli1$al[is.na(Sli1$al)] = mean_al
> Sli1$su[is.na(Sli1$su)] = mean_su
> Sli1$bgr[is.na(Sli1$bgr)] = mean_bgr
> Sli1$bu[is.na(Sli1$bu)] = mean_bu
> Sli1$bu[is.na(Sli1$bu)] = mean_bu
> Sli1$sd[is.na(Sli1$sd)] = mean_sd
> Sli1$sod[is.na(Sli1$sod)] = mean_sod
> Sli1$sc[is.na(Sli1$sc)] = mean_sc
> mean_pot <- as.integer(mean(Sli1$pot, na.rm = TRUE))
> Sli1$pot[is.na(Sli1$pot)] = mean_pot
> Sli1$hemo[is.na(Sli1$hemo)] = mean_hemo
> Sli1$pcv[is.na(Sli1$pcv)] = mean_pcv
> Sli1$wc[is.na(Sli1$wc)] = mean_wc
> Sli1$rc[is.na(Sli1$rc)] = mean_rc
```

Checking if value NA no longer exist.

```
> sum(is.na(Sli1))
[1] 0
```

There is no longer NA values that exist in the dataset.

D. Fixing the missing value in categorical columns . First, I replace all empty cell to NA and check how many data are missing in each column:

```
> Sli1$bp[Sli1$bp==""] <-NA
> sum(is.na(Sli1$bp))
[1] 0
> Sli1$rbc[Sli1$rbc==""] <-NA
> sum(is.na(Sli1$rbc))
[1] 142
> Sli1$pc[Sli1$pc==""] <-NA
> sum(is.na(Sli1$pc))
[1] 61
> Sli1$pcc[Sli1$pcc==""] <-NA
> sum(is.na(Sli1$pcc))
[1] 4
> Sli1$ba[Sli1$ba==""] <-NA
> sum(is.na(Sli1$ba))
[1] 4
```

```
> sum(is.na(Sli1$htn))
[1] 2
> Sli1$dm[Sli1$dm==""] <-NA
> sum(is.na(Sli1$dm))
[1] 2
> Sli1$cad[Sli1$cad==""] <-NA
> sum(is.na(Sli1$cad))
[1] 2
> Sli1$appet[Sli1$appet==""] <-NA
> sum(is.na(Sli1$appet))
[1] 1
> Sli1$pe[Sli1$pe==""] <-NA
> sum(is.na(Sli1$pe))
[1] 1
> Sli1$ane[Sli1$ane==""] <-NA
> sum(is.na(Sli1$ane))
[1] 1
> Sli1$classification[Sli1$classification==""] <-NA
> sum(is.na(Sli1$classification))
```

The result shows that there is a significant amount of missing observations from variable “rbc”, which has 142 (35%) and “pc” which has 61(15%). I rather exclude both variable as replacing them will make the data less trustworthy.

I use also count function and installed additional package/library like tidyverse, forcats to support further and enhance data pre-processing and integrate data visualization in one command.

```
> count(Sli1, rbc, sort = T)
       rbc   n
1   normal 198
2     <NA> 142
3 abnormal  45
> count(Sli1, pc, sort = T)
        pc   n
1   normal 253
2 abnormal  71
3     <NA>  61
> count(Sli1, pcc, sort = T)
         pcc   n
1 notpresent 340
2    present  41
3       <NA>   4
```

```
> count(Sli1, ba, sort = T)
          ba   n
1 notpresent 360
2    present  21
3       <NA>   4
> count(Sli1, htn, sort = T)
   htn   n
1   no 243
2  yes 140
3 <NA>   2
> count(Sli1, dm, sort = T)
    dm   n
1   no 254
2  yes 128
3 <NA>   2
4  yes   1
```

In “dm” variable there is 1 yest that is misspelled therefore it will be corrected.

```
> count(Sli1, cad, sort = T)
   cad   n
1   no 349
2  yes  34
3 <NA>   2
> count(Sli1, aooet, sort = T)
Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `aooet` is not found.
Run `rlang::last_error()` to see where the error occurred.
> count(Sli1, appet, sort = T)
  appet   n
1  good 304
2  poor  80
3  <NA>   1
```

```
> count(Sli1, pe, sort = T)
    pe   n
1   no 310
2  yes  74
3 <NA>   1
> count(Sli1, ane, sort = T)
   ane   n
1   no 325
2  yes  59
3 <NA>   1
> count(Sli1, classification, sort = T)
  classification   n
1            ckd 235
2         notckd 150
> detach("package:forcats", unload = TRUE)
> count(Sli1, classification, sort = T)
  classification   n
1            ckd 235
2         notckd 150
```

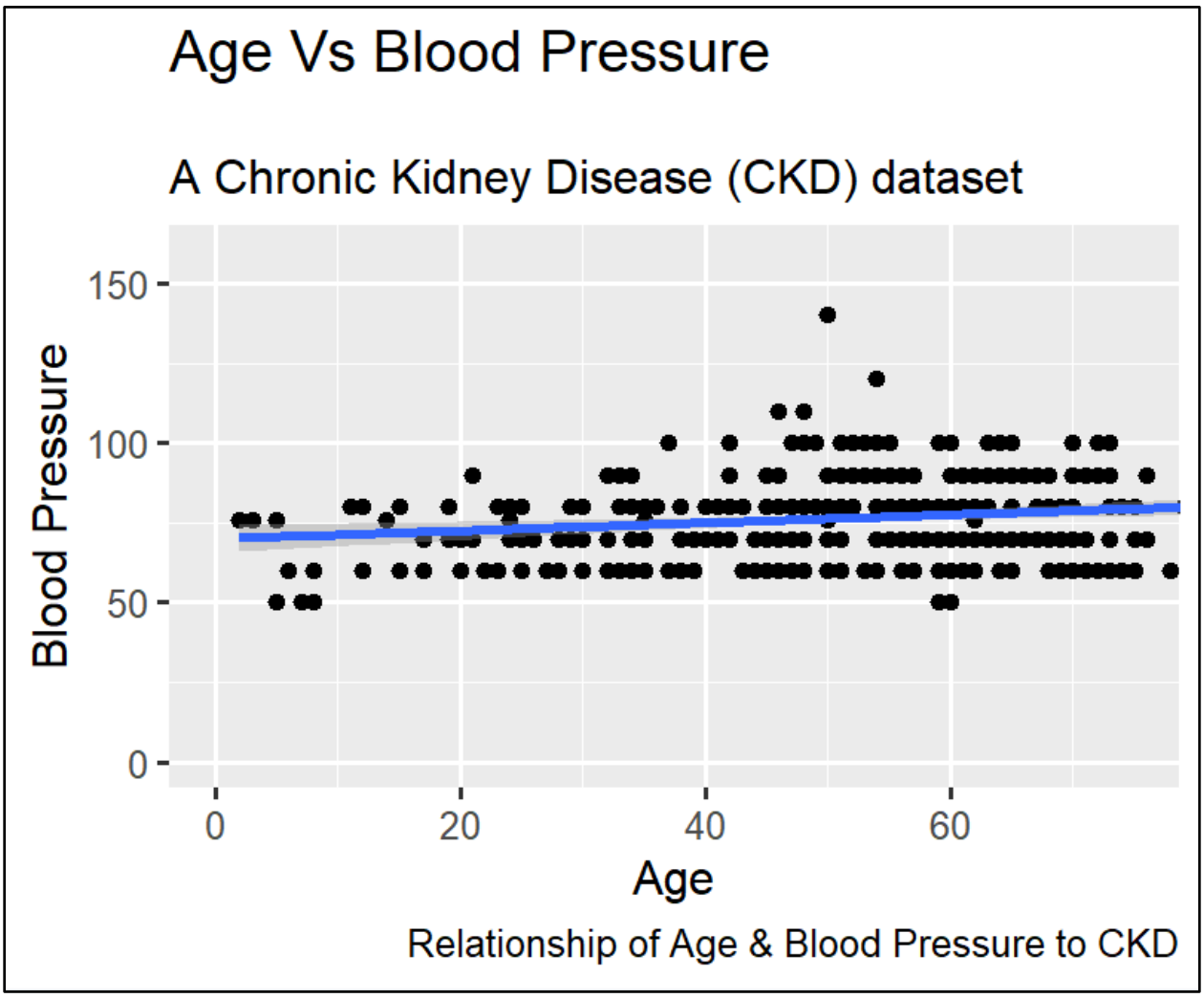
So far there are only two (rbc, pc) categorical variables that I will exclude in the analysis since there is not enough available data compared to the size of the observation. The rest have very minimal missing data and it will be dropped / deleted.

• bp - blood pressure • sg - specific gravity • al - albumin • su - sugar • rbc - red blood cells • pc - pus cell • pcc - pus cell clumps • ba - bacteria • bgr - blood glucose random • bu - blood urea • sc - serum creatinine • sod - sodium • pot - potassium • age - age	• Hemo- hemoglobin • pcv - packed cell volume • wc - white blood cell count • rc - red blood cell count • htn - hypertension • dm - diabetes mellitus • cad - coronary artery disease • appet - appetite • pe - pedal edema • ane - anemia • class - classification

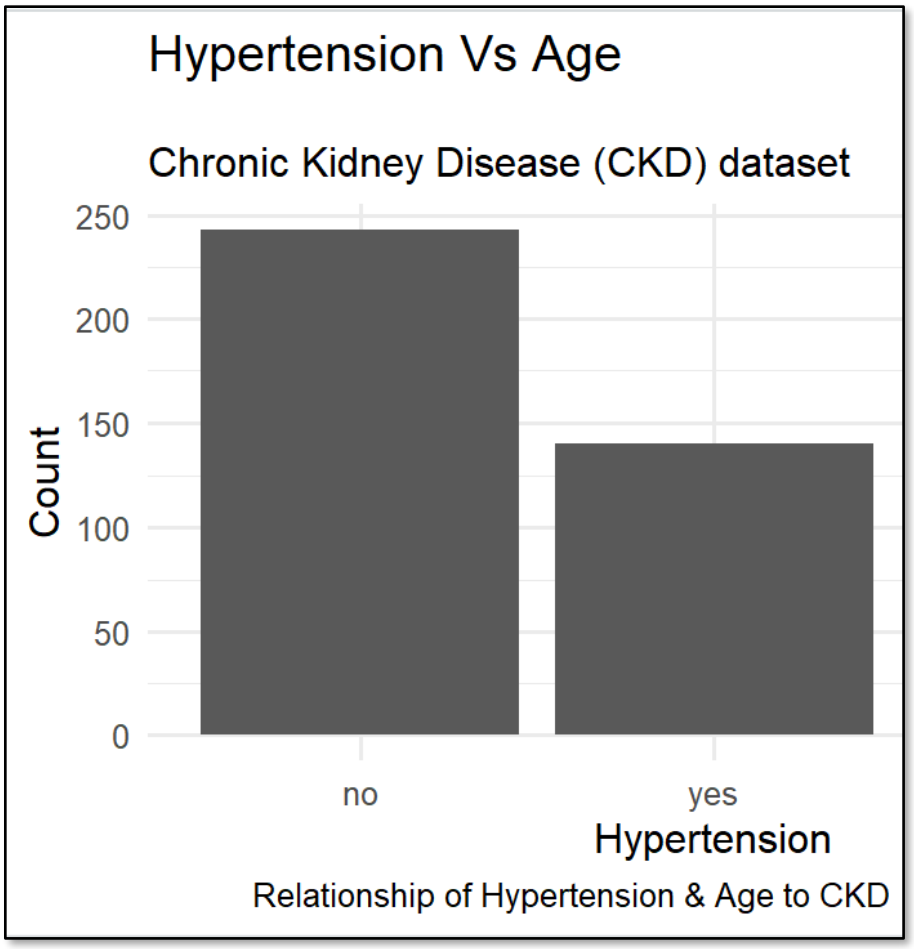
I want use variables Blood Pressure (bp), and Diabetes Mellitus (dm) vs Age (age) to represent in the Data Visualization.

Data Visualization.

I want use variables Blood Pressure (bp), and Diabetes Mellitus (dm) vs Age (age) to represent in the Data Visualization.




```
> g <- ggplot(dvisual, aes(x=age, y=bp)) +
+   geom_point() + geom_smooth(method="lm")+
+   coord_cartesian(xlim=c(0, 75), ylim=c(0, 160)) +
+   labs(title="Age Vs Blood Pressure", subtitle="
+ A Chronic Kidney Disease (CKD) dataset", y="Blood Pressure", x
="Age",
+        caption="Relationship of Age & Blood Pressure to CKD")
> plot(g)
```




```
> p<-ggplot(Sli1, aes(x=htn)) +
+   geom_bar()+
+   labs(title="Hypertension Vs Age", subtitle="
+ Chronic Kidney Disease (CKD) dataset", y="Count", x="Hypertension",
+        caption="Relationship of Hypertension & Age to CKD    Chronic Kidney Dis
ease")+
+   theme_minimal()
> p
```