1

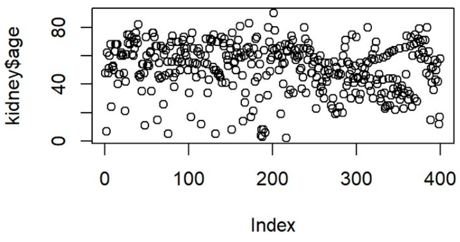
```
cities <- data.frame(ran <-runif (30,min=25,max=40))


i <- 1


while(i<10){
  ran <-runif (30,min=25,max=40)


  cities = cbind(cities, ran)


  i <- i+1
}
names(cities) <- c("c1","c2","c3","c4","c5","c6","c7","c8","c9","c10")


#average of every column, manual calculation
sum <- function (){
  sum <- colSums(cities)/30
  return(sum)
}
sum()
```

```
#manual function to get minimum number on every column.

  minvalue <- sort(cities$c1)
  min1 <- minvalue[1]
  minvalue <- sort(cities$c2)
  min2 <- minvalue[1]
  minvalue <- sort(cities$c3)
  min3 <- minvalue[1]
  minvalue <- sort(cities$c4)
  min4 <- minvalue[1]
  minvalue <- sort(cities$c5)
  min5 <- minvalue[1]
  minvalue <- sort(cities$c6)
  min6 <- minvalue[1]
  minvalue <- sort(cities$c7)
  min7 <- minvalue[1]
  minvalue <- sort(cities$c8)
  min8 <- minvalue[1]
  minvalue <- sort(cities$c9)
  min9 <- minvalue[1]
  minvalue <- sort(cities$c10)
  min10 <- minvalue[1]
  #sorting every column so the value of of the lowest temperature
  #is placing itself on position 1.
  mintot <- data.frame(min1,min2,min3,min4,min5,min6,min7,min8,min9,min10)
  #Printing the vectors of every position 1
View(mintot)
```

```
#manual function to get maximum number on every column.


maxvalue <- sort(cities$c1,decreasing = TRUE)
max1 <- maxvalue[1]
maxvalue <- sort(cities$c2,decreasing = TRUE)
max2 <- maxvalue[1]
maxvalue <- sort(cities$c3,decreasing = TRUE)
max3 <- maxvalue[1]
maxvalue <- sort(cities$c4,decreasing = TRUE)
max4 <- maxvalue[1]
maxvalue <- sort(cities$c5,decreasing = TRUE)
max5 <- maxvalue[1]
maxvalue <- sort(cities$c6,decreasing = TRUE)
max6 <- maxvalue[1]
maxvalue <- sort(cities$c7,decreasing = TRUE)
max7 <- maxvalue[1]
maxvalue <- sort(cities$c8,decreasing = TRUE)
max8 <- maxvalue[1]
maxvalue <- sort(cities$c9,decreasing = TRUE)
max9 <- maxvalue[1]
maxvalue <- sort(cities$c10,decreasing = TRUE)
max10 <- maxvalue[1]
#sorting every column backwards so the value of of the highest temperature
#is placing itself on position 1.
maxtot <- data.frame(max1,max2,max3,max4,max5,max6,max7,max8,max9,max10)
#Printing the vectors of every position 1
View(maxtot)



#inbuild function to get average of every column
for (i in colnames(cities)){
  print(mean(cities[[i]]))
}
#inbuild function to get minimum number on every column
minimum <- min(cities)
return(minimum)

#inbuild function to get maximum number on every column
maximum <- max(cities)
return(maximum)
```

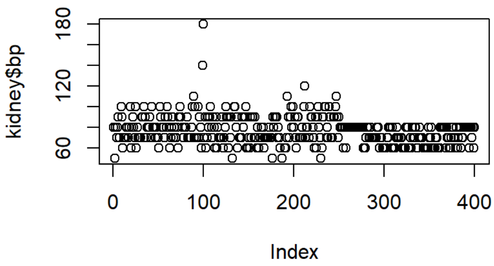
2

```
library(tidyverse)
kidney <- read.csv("C:/Data Scientist/R programmering/Final Assignment/kidney_disease.csv")
class(kidney)
#Checking the structure of data and coming up with the list of problems.
View(kidney)
str(kidney) #to see the setup and what needs to be changed. All the class
#for every column. These will be referred to in every column hereafter.
(sapply(kidney, unique))#gives unique values, to identify what is more than just
#an empty value, NA, or/and misspelled.
#Column “id” is a copy of the ruler of the y axis, with a slight deviation,
#since it starts at “0”. It will be removed.
kidney$id <- NULL #This removed the column “id”.
#Column “age”
#to see that there is 9 NA in this column.
#Plotting the “age” column shows that age on the participants is widespread and
#an average would be suited to #use in this case.
plot(kidney$age)
```



```
kidney$age[is.na(kidney$age)] <- (mean(kidney$age, na.rm = TRUE)) # to change
#all the NA to the #average(mean) number. The average is 51.4725, which means it
#has decimals that the rest of the column doesn’t have. #The column is also set
#to numeric. Since this is an average, it will comfort to the rest of the column
#and be forced to be a number without decimal.
kidney$age <- as.integer(kidney$age)
#Column“bp” have all its positions in the tenth of number. That is 20,30,40 and
#so on. It also have 12 NA out of 400 #rows. The plot shows a widespread of numbers.
#So a “mean” could replace the positions.
mean(kidney$bp[!is.na(kidney$bp)]) # the mean #calculated to 76.46907. This will
```

```
#be rounded up to 80 #for 6 of the positions and 70 for the other 6, so not to
#mess up then actual settings of whole tenth numbers.
```



```
kidney[8, 2] = 80
kidney[76, 2] = 80
kidney[133, 2] = 80
kidney[139, 2] = 80
kidney[162, 2] = 80
kidney[165, 2] = 80
#The rest is NA, so they will be changed to 70 with
kidney$bp[is.na(kidney$bp)] <- 70
#in “sg”, “al” and “su”, the NA, is almost all at the same rows. In “sg”,
#47 values are set to NA. Reading the Kaggle info, “sg” value is Specific Gravity
#and is the relative density between one substance and a reference substance.
#The range of this value is between 1,005 to 1,025. Without any reference point
#of the magnitude of these values they will not get an average value due to their
#slim variation and comparing to the index in the picture below, any deviation
#might give a less accurate plot. A KNN imputation would give a better estimated
#value. But #since a KNN is not available, these rows will be deleted so they won’t
#obscure the rest of the table.
which(is.na(kidney$sg)) # show all the NA in column “sg” to visually see how it
#correspond to “al” and “su”.
#They have their own NA that “sg” doesn’t. This will be dealt with separately.
kidney <- kidney [-c(14,18,22,29,31,38,51,58,60,79,82,83,87,99,100,105,107,110,
              122,123,126,133,143,149,152,161,163,167,175,189,198,202,203,
              204,206,209,223,229,232,237,239,246,269,281,296,323,347),]
#These rows are for “sg” and with the code above they were deleted.
rownames(kidney) <- 1:353# to have the accurate row numbers.
#“al” and “su” column is reset to integer
kidney$al <- as.integer(kidney$al)
kidney$su <- as.integer(kidney$su)
#The rows that “al” and “su” have NA separately is
```

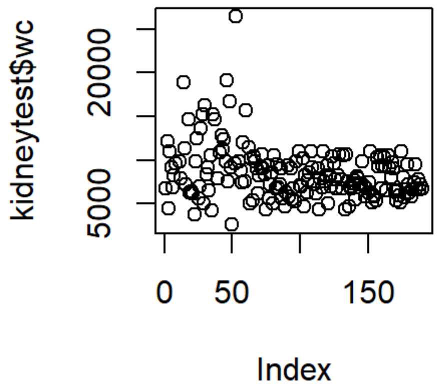
```
kidney$al[is.na(kidney$al)]
#Svar 2
#and
kidney$su[is.na(kidney$su)]
#Svar 4
#They will also be removed
which(is.na(kidney$al))
#Svar 113, 115
which(is.na(kidney$su))
#Svar 74, 113, 115, 165
kidney <- kidney [-c(74, 113, 115, 165),]
rownames(kidney) <- 1:349 #To reset the rownames.
#Column “rbc” is a chr, and it seem to have a lot of empty spaces. To see how many,
#kidneytest is created
kidneytest <- kidney
kidneytest$rbc [kidneytest$rbc == ''] <- NA# and all the empty spaces become NA.
kidneytest$rbc[is.na(kidneytest$rbc)] # then count the NA in kidneytest that
#actually accounts for the empty #values in kidney. 107 of the cells are empty.
#This column together with other columns farther down the row-lane has equally
#missing values and NA. This column will be untouched in this phase.
#With the same method as in “rbc”, column”pc” will make use of kidneytest since
#it has no NA and only empty values:
kidneytest$pc [kidneytest$pc == ''] <- NA
kidneytest$pc[is.na(kidneytest$pc)]
# the amount of NA in this column is 21. The issue here is that being chr,
#and the values are binary, a not answer could have a certain value. The empty
#spaces will be set to NA. The column will be changed to a factor with value 1
#for “normal”, 2 for “abnormal” and 3 for “NA”.
kidney$pc [kidney$pc == ''] <- NA
#The column “pc” is set to chr, so it has to be converted to a factor.
kidney$pc <- as.factor(kidney$pc)
```

```
kidney$pc <- factor(kidney$pc,levels = c("normal","abnormal","NA"))
#The column “pcc” is also a chr, and has the unique values of “present”,
#“notpresent” and empty spaces.
#To check how many spaces, kidneytest is used again.
kidneytest <- kidney
kidneytest$pcc[kidneytest$pcc == ''] <- NA
kidneytest$pcc[is.na(kidneytest$pcc)] # To count how many empty spaces kidney$pcc
#have, and its 4 of them.
#This column will be treated as the previous one, where the empty spaces will be
#filled with NA, and the factor will be set to 1 as “present”, 2 as “notpresent”
#and 3 as “NA”. These are the steps
kidney$pcc[kidney$pcc == ''] <- NA
kidney$pcc <- as.factor(kidney$pcc) # to make the column a factor.
kidney$pcc <- factor(kidney$pcc,levels = c("present","notpresent","NA"))
#The column “ba” is also a chr. I have the same setup as “pcc”, with only 4 empty
#spaces. So, the same application will be applied to “ba” as in “pcc”. The steps
#are summarized below.
kidneytest$ba[kidneytest$ba==''] <- NA
kidneytest$ba[is.na(kidneytest$ba)] #This was the test to determine the amount of NA.
unique(kidney$ba) # to determinate the different value names. Those are “present”,
#“notpresent” and empty value.
kidney$ba[kidney$ba == ''] <- NA
kidney$ba <- as.factor(kidney$ba)
kidney$ba <- factor(kidney$ba,levels = c("present","notpresent","NA"))
#Column “bgr” is numeric with NA values.
which(is.na(kidney$bgr)) # to show that bgr have 38 NA values. This is more than
#1/10 of all the values since rows has been deleted and is down to 349. Using the
#kidneytest dataframe to see how many NA were in bgr from the start gives
which(is.na(kidneytest$bgr)) # 40, so it is still 1/10 of the rows without a value.
#The initial rows were 400.
#It would be difficult to create an average to mimic the rest of the values.
```

```
#This is sort of trend for the 5 columns after, “bu”, “c”, ”sod”, “pot” and “hemo”.
#They are all numeric, and they all have a large chunk of NA.
which(is.na(kidney$bu)) # with 18 NA.
which(is.na(kidney$sc)) # with 17 NA.
which(is.na(kidney$sod)) # with 75 NA.
which(is.na(kidney$pot)) # with 75 NA.
which(is.na(kidney$hemo)) # with 39 NA.
# The NA together with the kidneytest of “rbc”, that have 107 NA, there are similar
#rows. Doing a visual check, it seems to correspond well. Having these nonvalues
#in the dataset will only obscure the registered values.
#The rows of “rbc” in test version will be deleted first to make this estimation more solid.
kidneytest$rbc[kidneytest$rbc==''] <- NA # to rbc empty values to NA
which(is.na(kidneytest$rbc)) # These are the rows that will be deleted.
kidneytest <- kidneytest [-c(1,2,6,7,11,13,15,16,17,18,21,23,24,27,28,36,37,40,
                        41,42,45,46,47,52,54,56,57,60,61,62,64,67,69,70,71,
                        75,76,81,82,83,84,86,87,89,91,92,93,95,97,99,100,
                        101,102,103,104,105,108,110,112,114,115,116,125,129,
                        134,135,136,138,139,140,142,144,148,151,152,153,154,
                        160,165,166,168,169,170,172,175,177,178,179,180,181,
                        183,184,187,193,194,196,197,199,200,243,261,299,300,
                        331),]
rownames(kidneytest) <- 1:245 # This is done in kidneytest to see if the the actual
#similarities to the other colums
which(is.na(kidneytest$bu)) # with 12 NA.
which(is.na(kidneytest$sc)) # with 11 NA.
which(is.na(kidneytest$sod)) # with 34 NA.
which(is.na(kidneytest$pot)) # with 34 NA.
which(is.na(kidneytest$hemo)) #with 22 NA.
# So the corresponding NA to the column “bgr” is almost half compared. Had this
#been a lower number this wouldn’t have been significant, but somehow it has a
#connection. So the test will be made real on kidney dataset, and the rows for
```

```
#“bu”, “sc”, “sod”, “pot” and “hemo” will be deleted as well.
kidney$rbc[kidney$rbc==''] <- NA
kidney <- kidney [-c(1,2,6,7,11,13,15,16,17,18,21,23,24,27,28,36,37,40,41,42,45,
              46,47,52,54,56,57,60,61,62,64,67,69,70,71,75,76,81,82,83,84,
              86,87,89,91,92,93,95,97,99,100,101,102,103,104,105,108,110,
              112,114,115,116,125,129,134,135,136,138,139,140,142,144,148,
              151,152,153,154,160,165,166,168,169,170,172,175,177,178,179,
              180,181,183,184,187,193,194,196,197,199,200,243,261,299,300,
              331),]
rownames(kidney) <- 1:245 # kidney is now the same as the test.
#The remaining in column “rbc” will be changed to the class factor and the
unique(kidney$rbc) # shows the only 2 values left are “normal” and abnormal
#The factors will be set as 1 as “normal and 2 as ”abnormal”.
kidney$rbc <- as.factor(kidney$rbc)
kidney$rbc <- factor(kidney$rbc,levels=c("normal","abnormal"))
#Column “bu”
which(is.na(kidney$bu)) # to have the rows to get deleted.
kidney <- kidney[-c(25,26,32,69,91,97,131,137,163,166,184,225),]
rownames(kidney) <- 1:233
#Column “sc”
which(is.na(kidney$sc)) # This is now 0 NA.
#Column “sod”
which(is.na(kidney$sod))
kidney <- kidney[-c(1,3,4,5,13,14,15,16,20,24,36,42,44,45,46,58,59,62,66,70,72,
              75,93,147,175,199),]
rownames(kidney) <- 1:207 # The numbers of NA, is shrinking steady, so the NA for
#the 5 column seem to connect on some level.
#Column “pot” has no 0 NA
which(is.na(kidney$pot))
#Column “hemo”
which(is.na(kidney$hemo))
```

```
kidney <- kidney[-c(17,19,24,30,38,41,53,65,69,70,74,99,136,140,144,146,175),]
rownames(kidney) <- 1:190
#Column “pcv” has no NA, but there some values missing. It is also a chr, but
#with numbers.
(unique(kidney$pcv)) # shows that there are one position that is missing a value.
#This position will be replaced by an average.
kidney$pcv[kidney$pcv==""] <- NA # The blank space is first changed to NA
kidney$pcv <- as.integer(kidney$pcv) # the class is changed to integer so we can
#perform a mean function.
kidney$pcv[is.na(kidney$pcv)] <- (mean(kidney$pcv, na.rm = TRUE)) # The NA is now
#the mean. The problem is that the mean is a decimal value. So, even though it is
#an integer, it shows decimal. The new value is forced into no decimal value with
#another
kidney$pcv <- as.integer(kidney$pcv)
#Column “wc” is shown to have errors in spacing, that is \t before a value, and
#empty values shown from
unique(kidney$wc)
kidney$wc [is.na(kidney$wc)] # it has 0 NA
#Doing a kidneytest to convert the empty spaces in “wc” to NA to see how many give us
kidneytest <- kidney
kidneytest$wc[kidneytest$wc ==""] <- NA
kidneytest$wc [is.na(kidneytest$wc)] # There are 12 NA.
```



```
#Plotting this test to see the variation, one can determinate that it has a spread
#and a mean to the blank spaces would be preferable.
#But first this needs to be changed to an integer since it is set to chr.
kidney$wc <- as.integer(kidney$wc)
#mean(kidney$wc, na.rm = TRUE)) # The average value of wc is 8457.865. This will
#be rounded not to make unique values when plotting, to suit the style of other
#values in “wc” = 8500
kidney$wc[is.na(kidney$wc)] <- 8500
#Column”rc”
#Using kidneytest to determinate how many NA the column of “rc” have gives:
kidneytest <- kidney
kidneytest$rc [kidneytest$rc==''] <- NA
kidneytest$rc [is.na(kidneytest$rc)] # There are 16 blank spaces in kidney$rc
#kidney$rc is set to be chr, but will first changer to integer.
kidney$rc <- as.integer(kidney$rc)
#The mean will value will be used on these 16 blank spaces.
kidney$rc [kidney$rc==''] <- NA
#mean(kidney$rc, na.rm = TRUE)) # This equals 4.385057, but will be rounded off
#to 4 to fit the other values in column”rc”.
kidney$rc[is.na(kidney$rc)] <- 4
#Column “htn”, “dm” and “cad”
```

```
#Kindneytest will be used to determinte how many blankspaces on all three.
#A visual look have them set to have them on the same rows.
#“htn”
kidneytest <- kidney
kidneytest$htn [kidneytest$htn==''] <- NA
kidneytest$htn [is.na(kidneytest$htn)] # 2 blank spaces.
#“dm”
kidneytest <- kidney
kidneytest$dm [kidneytest$dm==''] <- NA
kidneytest$dm [is.na(kidneytest$dm)] # 2 blank spaces.
#“cad”
kidneytest$cad [kidneytest$cad==''] <- NA
kidneytest$cad [is.na(kidneytest$cad)] # 2 blank spaces.
#Yes, they were on the same row and with only 2 blank spaces.
#Since the values are yes and no, the blank spaces will only be changed to NA
#and not given a mean value. The columns will have it class changed to factor
#and set a value of 1 to “yes”, 2 to “no” and 3 to “NA”.
#“htn”
kidney$htn <- as.factor(kidney$htn)
kidney$htn [kidney$htn==''] <- NA
kidney$htn <- factor(kidney$htn,levels=c("yes","no","NA"))
#“dm”
kidney$dm <- as.factor(kidney$dm)
kidney$dm [kidney$dm==''] <- NA
kidney$dm <- factor(kidney$dm,levels=c("yes","no","NA"))
#“cad”
kidney$cad <- as.factor(kidney$cad)
kidney$cad [kidney$cad==''] <- NA
kidney$cad <- factor(kidney$cad,levels=c("yes","no","NA"))
#In the next 3 columns, “appet”, “pe” and ane
#It is the same situation as the previous 3 columns.
```

```
#This is shown by doing the kidneytest on the 3 columns

#“appet”

kidneytest <- kidney

kidneytest$appet [kidneytest$appet=='']  <- NA

kidneytest$appet [is.na(kidneytest$appet)] # 1 blank spaces.

#“pe”

kidneytest <- kidney

kidneytest$pe [kidneytest$pe=='']  <- NA

kidneytest$pe [is.na(kidneytest$pe)] # 2 blank spaces.

#“ane”

kidneytest$ane [kidneytest$ane=='']  <- NA

kidneytest$ane [is.na(kidneytest$ane)] # 2 blank spaces.

#As in “htn”,”dm” and “cad”, these columns will be treated the same on the row

#with no values. Except for “appet” column that will have 1 as “good”, 2 as “poor”

#and 3 as “NA”.

#“appet”

kidney$appet <- as.factor(kidney$appet)

kidney$appet [kidney$appet=='']  <- NA

kidney$appet <- factor(kidney$appet,levels=c("good","poor","NA"))

#“pe”

kidney$pe <- as.factor(kidney$pe)

kidney$pe [kidney$pe=='']  <- NA

kidney$pe <- factor(kidney$pe,levels=c("yes","no","NA"))

#“ane”

kidney$ane <- as.factor(kidney$ane)

kidney$ane [kidney$ane=='']  <- NA

kidney$ane <- factor(kidney$ane,levels=c("yes","no","NA"))

#The last column is classification”

#With sapply(kidney$classification, unique) it can be seen to have no blank spaces,

#and 2 values of “ckd” and “notckd”. This column will be changed from chr to a

#factor with 2 levels. 1 will be “ckd” and 2 will be “notckd”.
```

```
kidney$classification <- as.factor(kidney$classification)

kidney$classification <- factor(kidney$classification,levels=c("ckd","notckd"))
```

3

Explanation to every vector name in the data frame Kidney.

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 = haemoglobin

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

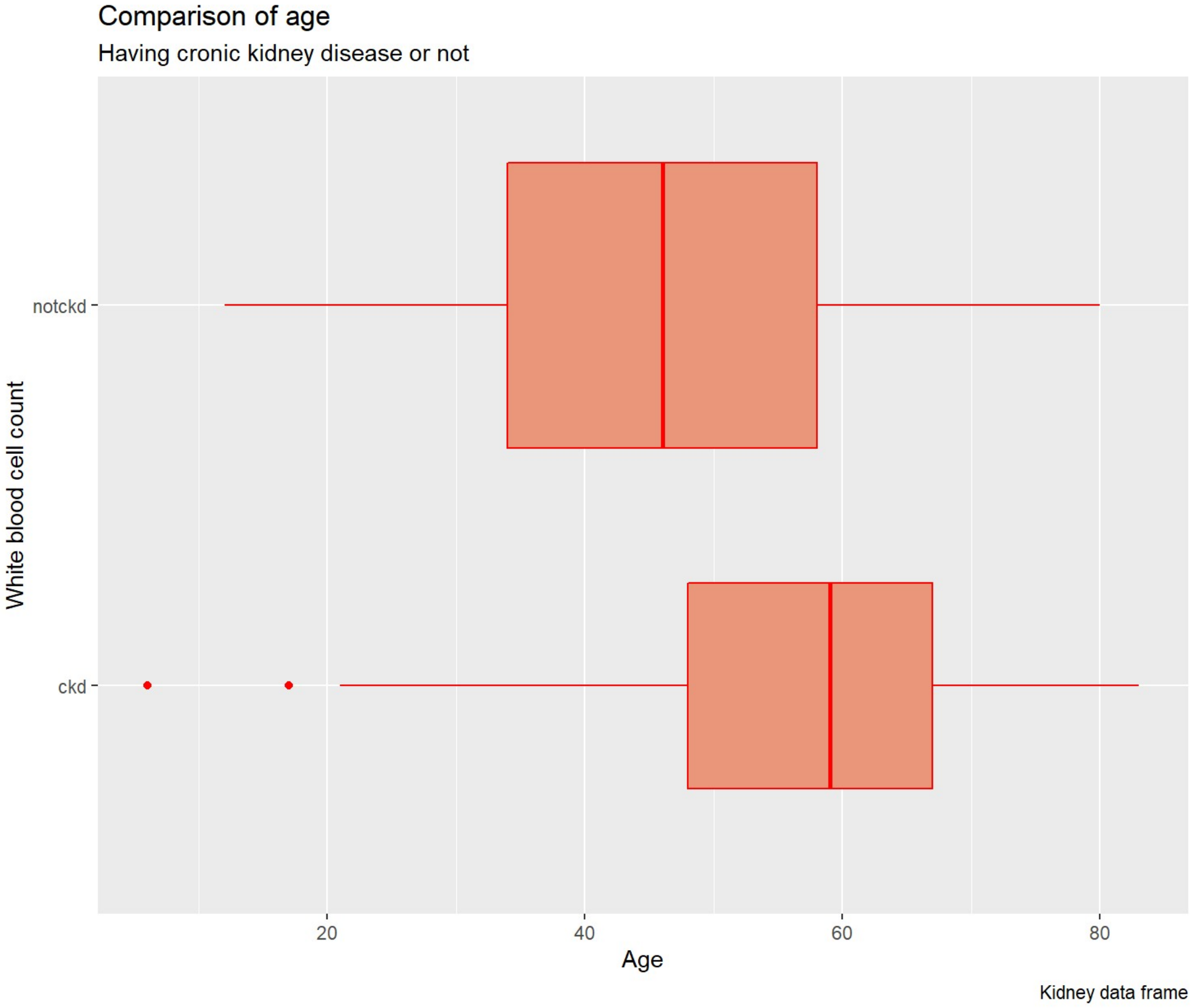
pe = peda_edema

ane = anemia

classification = Chronic Kidney Disease

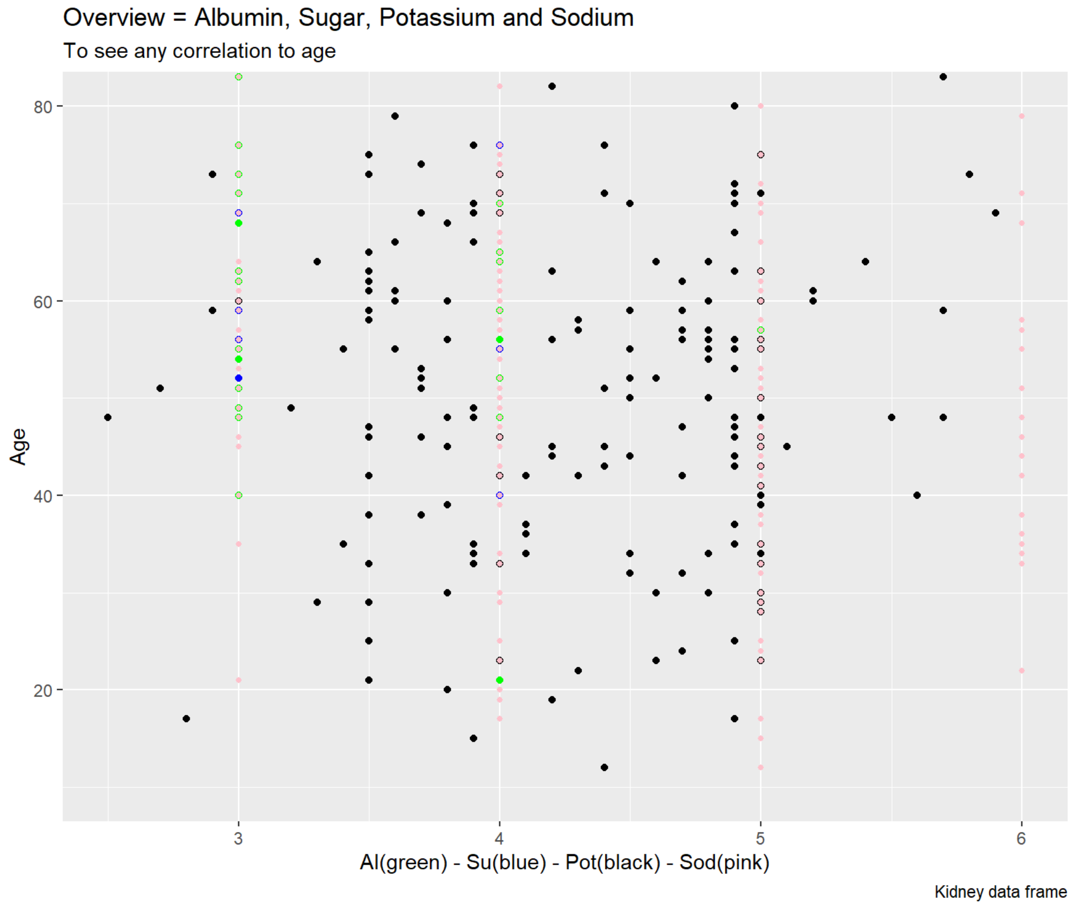
This is a boxplot comparison graph, with that shows the the median number in red. The dots outside the boxes are extreme points.

```
library(ggplot2)

theme_set(theme_gray())

g <- ggplot(kidney, aes(age,classification))

g + geom_boxplot(varwidth=T, fill="darksalmon",colour="red") +

  #warwidth=T for the boxes to be adjusted to the amount observations

  #in the boxes.

  labs(title="Comparison of age ",

       subtitle="Having cronic kidney disease or not ",

       caption="Kidney data frame",

       x="Age",

       y="White blood cell count")
```



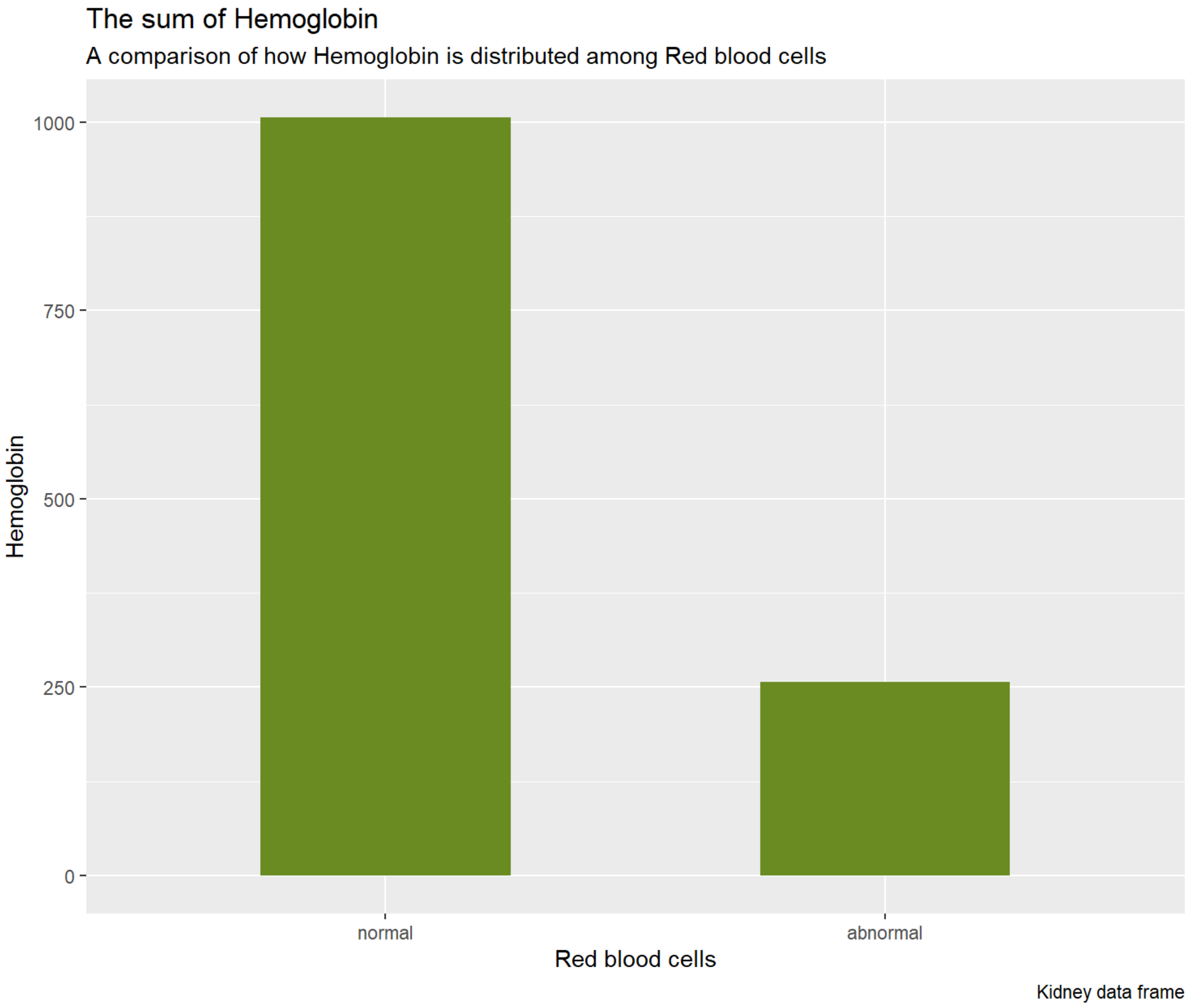
This is an attempt to visualize either the difference or the similarity of a compound, 2 elements and a protein.

```
library(ggplot2)

ggplot(kidney, aes(y=age)) +

  geom_point(aes(x = al), color = "green")+

  geom_point(aes(x = su), color="blue")+

  geom_point(aes(x = pot), color="black")+

  geom_point(aes(x = rc),size=1, color="pink")+

  #adding 4 layers of vector to visualised with different color.

  coord_cartesian(ylim=c(10, 80), xlim=c(2.5, 6))+ #zooming in.

  labs(title="Overview = Albumin, Sugar, Potassium and Sodium",

       subtitle="To see any correlation to age",

       caption = "Kidney data frame",

       y= "Age",

       x= "Al(green) - Su(blue) - Pot(black) - Sod(pink)")

#last part is labeling the title, subtitle, caption, y and x.
```



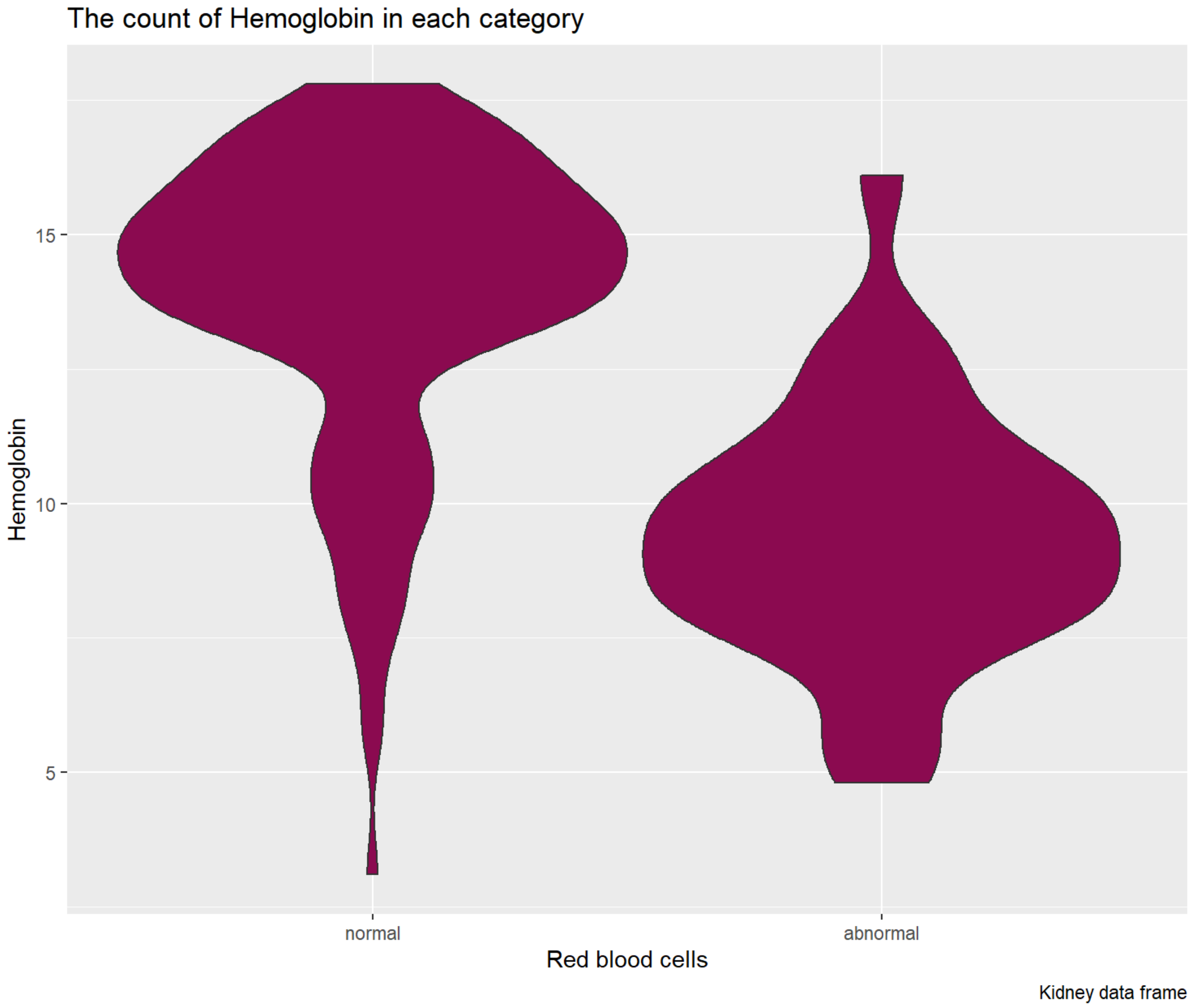
The amount of Hemoglobin in people that have normal red blood cells compared the abnormal.

```
library(ggplot2)

theme_set(theme_gray())


ggplot(kidney, aes(x=rbc, y=hemo)) +

  geom_bar(stat="unique", width=0.5, fill="olivedrab4") +

  labs(title="The sum of Hemoglobin",

  subtitle = "A comparison of how Hemoglobin is distributed among Red blood cells",

  caption="Kidney data frame",

  x="Red blood cells",y="Hemoglobin")
```



This is a continue of the last graph and shows the amount of people having the similar value of hemogoblin.

```
library(ggplot2)

theme_set(theme_gray()) # adding grey backround for clarity.


ggplot (kidney,aes(x=rbc,y=hemo))+

geom_violin(fill="deeppink4",width=1)+

#width 1 for clearer visualisation.

 labs(title="The count of Hemoglobin in each category",

    caption="Kidney data frame",

    x="Red blood cells",y="Hemoglobin")
```

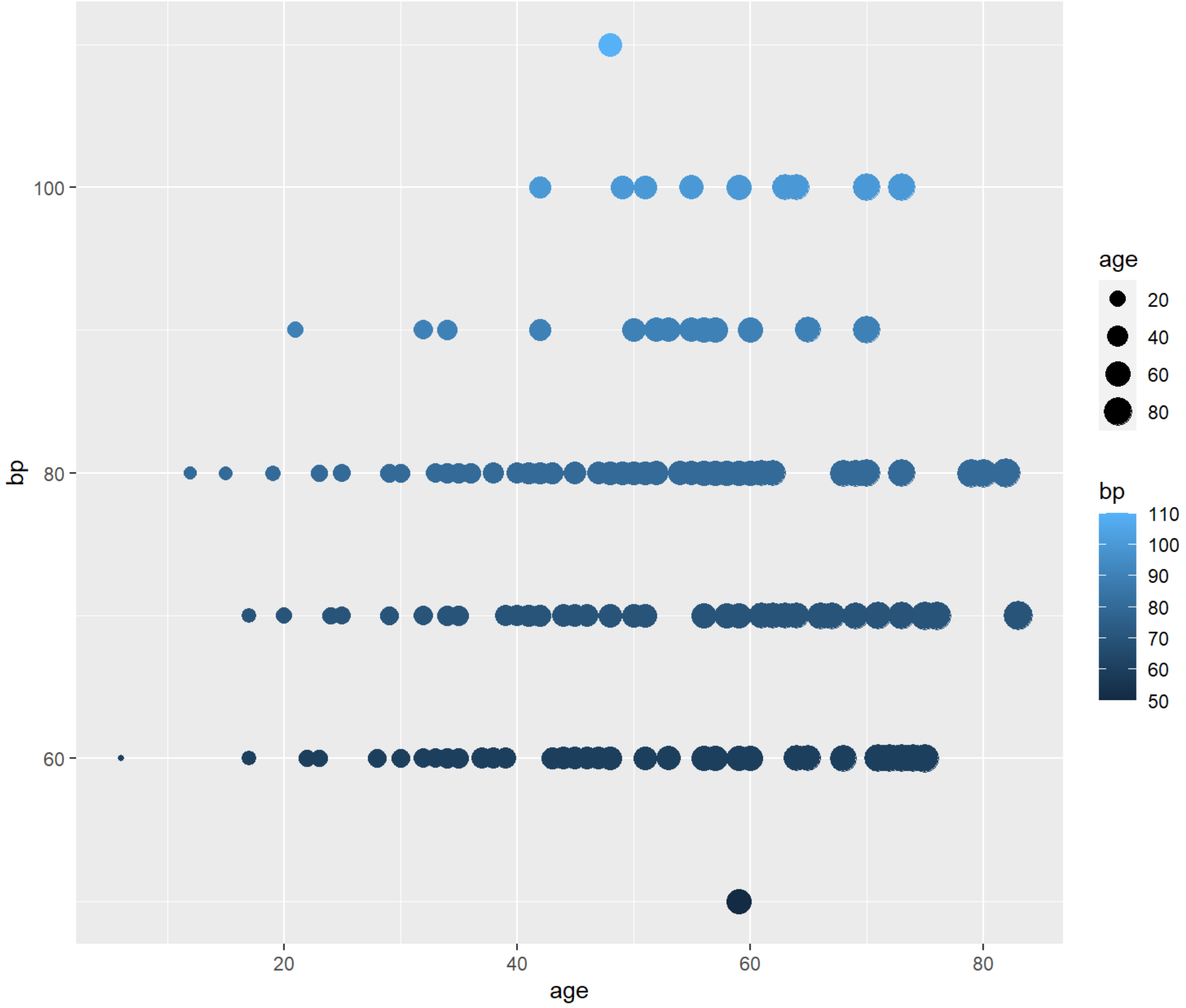


bp = blood pressure. The graph shows blood pressure according to age.

qplot(age, bp, data = kidney,colour=bp,size=age)

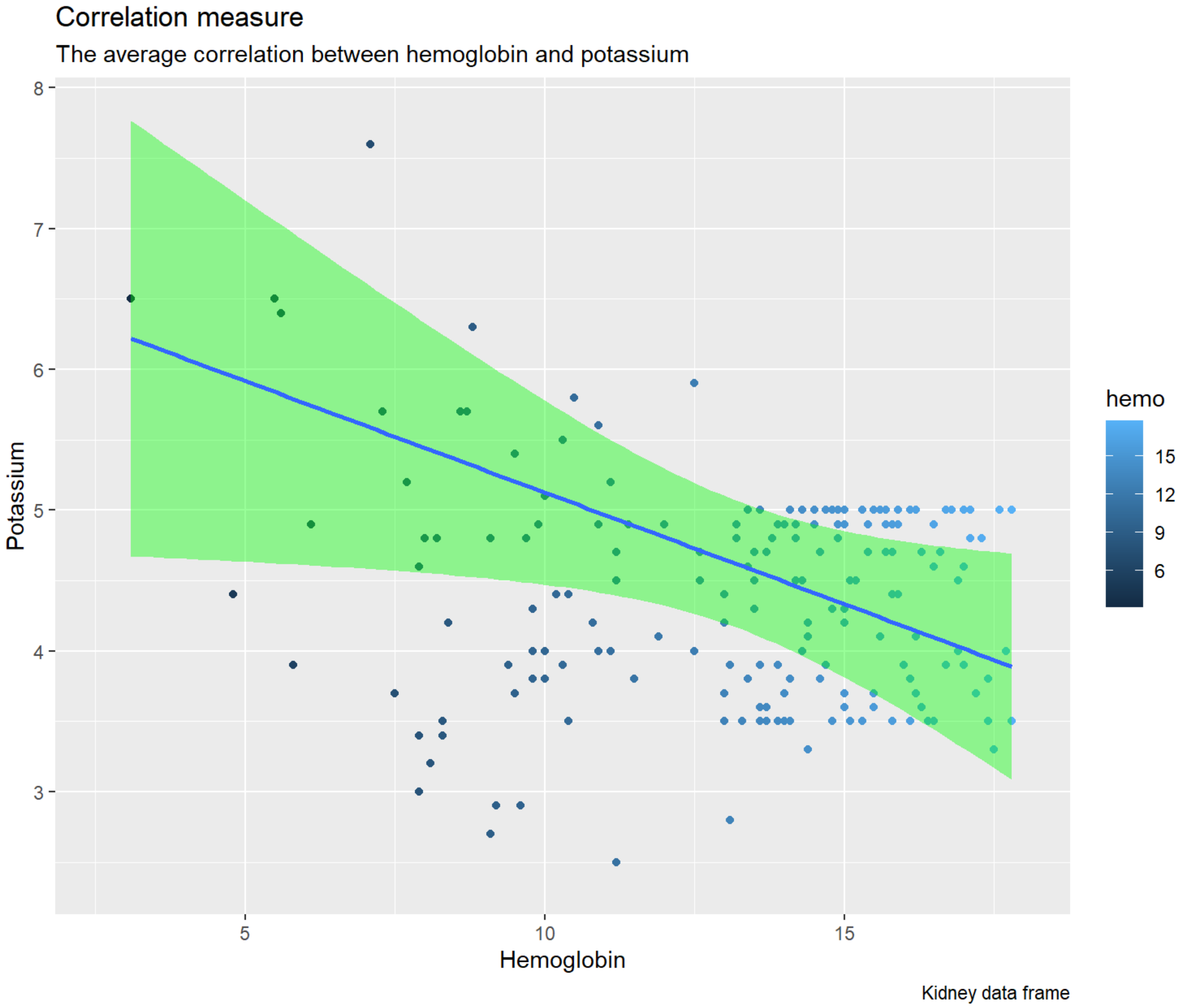
#using the dataframe kidney, colour is set according to

#one of the parameter and size is set to the other.



This is a very zoomed in graph of potassium is getting lower the higher the value of hemoglobin is.

```
library(ggplot2)
g6 <- ggplot(kidney, aes(x=hemo, y=pot,colour=hemo)) +
  geom_point()+
  geom_smooth(method = "lm",fill="green")
#smooth emphasize the lower frequency behaviour.
#lm to make a line through the field to zoom in.
g6+coord_cartesian(ylim=c(2.4, 7.8), xlim=c(2.6,18))+
  labs(title="Correlation measure",
       subtitle="The average correlation between hemoglobin and potassium",
       caption="Kidney data frame",
       x="Hemoglobin",      y="Potassium")
```


Correlation measure
The average correlation between hemoglobin and potassium
Potassium
Hemoglobin
hemo
Kidney data frame