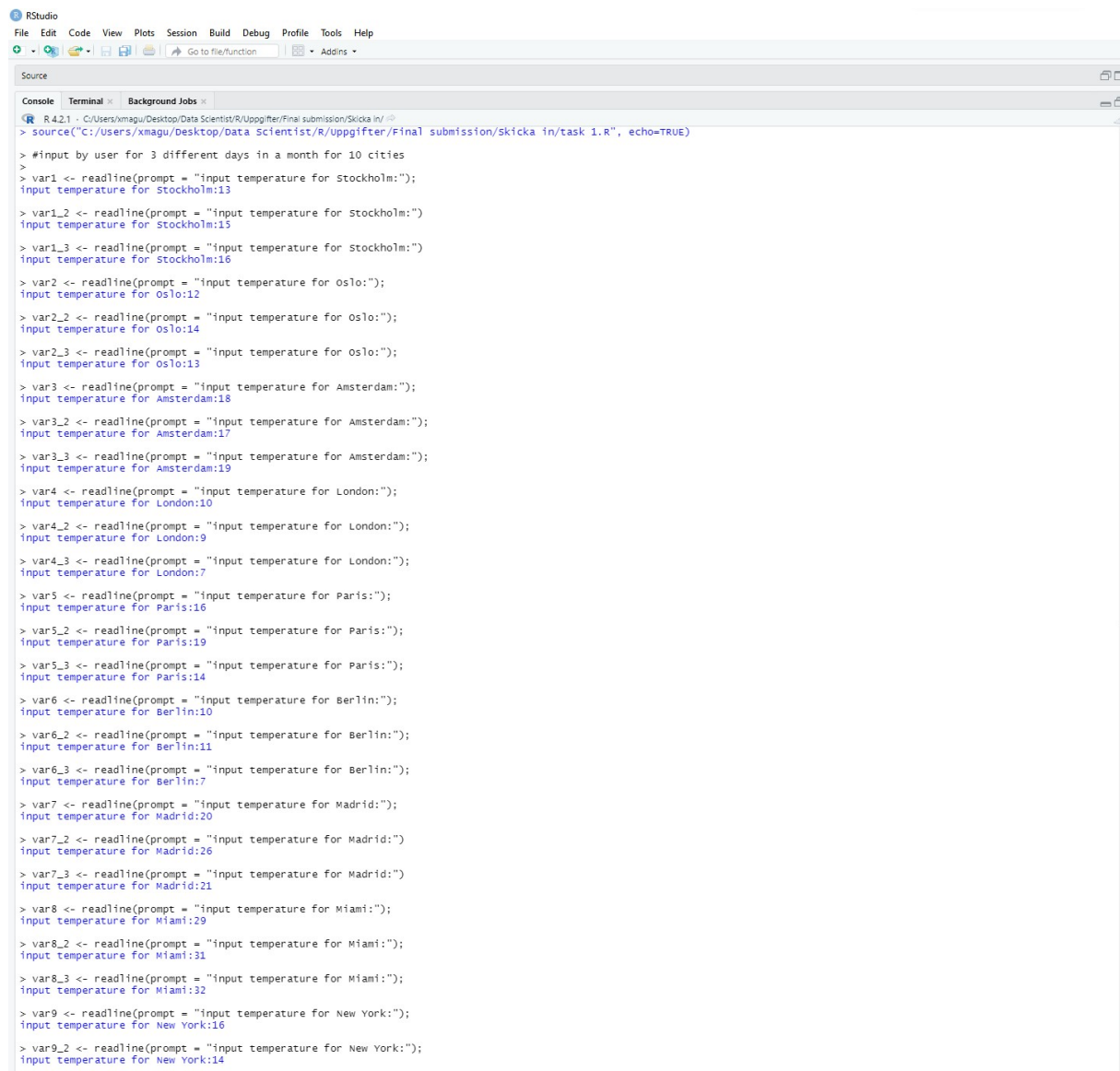


TASK 1) Here I input temperature for each city 3 times



```
R 4.2.1 - C:/Users/xmagu/Desktop/Data Scientist/R/Uppgifter/Final submission/Skicka in/
> source("C:/Users/xmagu/Desktop/Data Scientist/R/Uppgifter/Final submission/Skicka in/task 1.R", echo=TRUE)

> #input by user for 3 different days in a month for 10 cities
>
> var1 <- readline(prompt = "input temperature for Stockholm:");
input temperature for Stockholm:13

> var1_2 <- readline(prompt = "input temperature for Stockholm:");
input temperature for Stockholm:15

> var1_3 <- readline(prompt = "input temperature for Stockholm:");
input temperature for Stockholm:16

> var2 <- readline(prompt = "input temperature for Oslo:");
input temperature for Oslo:12

> var2_2 <- readline(prompt = "input temperature for Oslo:");
input temperature for Oslo:14

> var2_3 <- readline(prompt = "input temperature for Oslo:");
input temperature for Oslo:13

> var3 <- readline(prompt = "input temperature for Amsterdam:");
input temperature for Amsterdam:18

> var3_2 <- readline(prompt = "input temperature for Amsterdam:");
input temperature for Amsterdam:17

> var3_3 <- readline(prompt = "input temperature for Amsterdam:");
input temperature for Amsterdam:19

> var4 <- readline(prompt = "input temperature for London:");
input temperature for London:10

> var4_2 <- readline(prompt = "input temperature for London:");
input temperature for London:9

> var4_3 <- readline(prompt = "input temperature for London:");
input temperature for London:7

> var5 <- readline(prompt = "input temperature for Paris:");
input temperature for Paris:16

> var5_2 <- readline(prompt = "input temperature for Paris:");
input temperature for Paris:19

> var5_3 <- readline(prompt = "input temperature for Paris:");
input temperature for Paris:14

> var6 <- readline(prompt = "input temperature for Berlin:");
input temperature for Berlin:10

> var6_2 <- readline(prompt = "input temperature for Berlin:");
input temperature for Berlin:11

> var6_3 <- readline(prompt = "input temperature for Berlin:");
input temperature for Berlin:7

> var7 <- readline(prompt = "input temperature for Madrid:");
input temperature for Madrid:20

> var7_2 <- readline(prompt = "input temperature for Madrid:");
input temperature for Madrid:26

> var7_3 <- readline(prompt = "input temperature for Madrid:");
input temperature for Madrid:21

> var8 <- readline(prompt = "input temperature for Miami:");
input temperature for Miami:29

> var8_2 <- readline(prompt = "input temperature for Miami:");
input temperature for Miami:31

> var8_3 <- readline(prompt = "input temperature for Miami:");
input temperature for Miami:32

> var9 <- readline(prompt = "input temperature for New York:");
input temperature for New York:16

> var9_2 <- readline(prompt = "input temperature for New York:");
input temperature for New York:14

> var9_3 <- readline(prompt = "input temperature for New York:");
input temperature for New York:15

> var10 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:30

> var10_2 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:35

> var10_3 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:39
```

```
> var9 <- readline(prompt = "input temperature for New York:");
input temperature for New York:16

> var9_2 <- readline(prompt = "input temperature for New York:");
input temperature for New York:14

> var9_3 <- readline(prompt = "input temperature for New York:");
input temperature for New York:15

> var10 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:30

> var10_2 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:35

> var10_3 <- readline(prompt = "input temperature for Los Angeles:");
input temperature for Los Angeles:39
```

Part a)

By using manual calculations

```
> #Part A) By using manual calculations did this only with Stockholm
> #if else if else statement to see witch of them 3 is minimum the prints if th .... [TRUNCATED]
13 Minimum Temperature for Stockholm

> #if else if else Maximum to see witch of them 3 is minimum the prints if the statement filed
>
> if (var1 > var1_2 && var1 > var1_3) {
+   cat(var1 .... [TRUNCATED]
16 Maximum Temperature for Stockholm

> #Average
> b <- 3

> avg = a/b

> print(paste("average Temperature for Stockhom\n:", avg))
[1] "average Temperature for Stockhom\n: 4.33333333333333" "average Temperature for Stockhom\n: 5" "average Temperature for Stockhom\n: 5.33333333333333"
```

Part b) By using the in-built functions in R

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

Source
Console Terminal Background Jobs
R 4.2.1 - C:/Users/kmagu/Desktop/Data Scientist/R/Uppgifter/Final submission/Skicka inf
> # Part b) By using the in-built functions in R
>
> print(paste("Minimum Temperature for Stockholm:", min(a)))
[1] "Minimum Temperature for Stockholm: 13"
> print(paste("Maximum Temperature for Stockholm:", max(a)))
[1] "Maximum Temperature for Stockholm: 16"
> print(paste("average Temperature for Stockholm:", mean(a)))
[1] "average Temperature for Stockholm: 14.6666666666667"
> print(paste("Minimum Temperature for oslo:", min(b)))
[1] "Minimum Temperature for oslo: 3"
> print(paste("Maximum Temperature for oslo:", max(b)))
[1] "Maximum Temperature for oslo: 3"
> print(paste("average Temperature for oslo:", mean(b)))
[1] "average Temperature for oslo: 3"
> print(paste("Minimum Temperature for Amsterdam:", min(c)))
[1] "Minimum Temperature for Amsterdam: 17"
> print(paste("Maximum Temperature for Amsterdam:", max(c)))
[1] "Maximum Temperature for Amsterdam: 19"
> print(paste("average Temperature for Amsterdam:", mean(c)))
[1] "average Temperature for Amsterdam: 18"
> print(paste("Minimum Temperature for London:", min(d)))
[1] "Minimum Temperature for London: 7"
> print(paste("Maximum Temperature for London:", max(d)))
[1] "Maximum Temperature for London: 10"
> print(paste("average Temperature for London:", mean(d)))
[1] "average Temperature for London: 8.6666666666667"
> print(paste("Minimum Temperature for Paris:", min(e)))
[1] "Minimum Temperature for Paris: 14"
> print(paste("Maximum Temperature for Paris:", max(e)))
[1] "Maximum Temperature for Paris: 19"
> print(paste("average Temperature for Paris:", mean(e)))
[1] "average Temperature for Paris: 16.3333333333333"
> print(paste("Minimum Temperature for Berlin:", min(f)))
[1] "Minimum Temperature for Berlin: 7"
> print(paste("Maximum Temperature for Berlin:", max(f)))
[1] "Maximum Temperature for Berlin: 11"
> print(paste("average Temperature for Berlin:", mean(f)))
[1] "average Temperature for Berlin: 9.3333333333333"
> print(paste("Minimum Temperature for Madrid:", min(g)))
[1] "Minimum Temperature for Madrid: 20"
> print(paste("Maximum Temperature for Madrid:", max(g)))
[1] "Maximum Temperature for Madrid: 26"
> print(paste("average Temperature for Madrid:", mean(g)))
[1] "average Temperature for Madrid: 22.3333333333333"
> print(paste("Minimum Temperature for Miami:", min(h)))
[1] "Minimum Temperature for Miami: 29"
> print(paste("Maximum Temperature for Miami:", max(h)))
[1] "Maximum Temperature for Miami: 32"
> print(paste("average Temperature for Miami:", mean(h)))
[1] "average Temperature for Miami: 30.6666666666667"
> print(paste("Minimum Temperature for New York:", min(j)))
[1] "Minimum Temperature for New York: 14"
> print(paste("Maximum Temperature for New York:", max(j)))
[1] "Maximum Temperature for New York: 16"

> print(paste("average Temperature for New York:", mean(j)))
[1] "average Temperature for New York: 15"

> print(paste("Minimum Temperature for Los Angeles:", min(k)))
[1] "Minimum Temperature for Los Angeles: 30"

> print(paste("Maximum Temperature for Los Angeles:", max(k)))
[1] "Maximum Temperature for Los Angeles: 39"

> print(paste("average Temperature for Los Angeles:", mean(k)))
[1] "average Temperature for Los Angeles: 34.6666666666667"
```

#THE CODE FROM R STUDIO SCRIPT

#input by user for 3 different days in a month for 10 cities

```
var1 <- readline(prompt = "input temperature for Stockholm:");
```

```
var1_2 <- readline(prompt = "input temperature for Stockholm:")
var1_3 <- readline(prompt = "input temperature for Stockholm:")
var2 <- readline(prompt = "input temperature for Oslo:");
var2_2 <- readline(prompt = "input temperature for Oslo:");
var2_3 <- readline(prompt = "input temperature for Oslo:");
var3 <- readline(prompt = "input temperature for Amsterdam:");
var3_2 <- readline(prompt = "input temperature for Amsterdam:");
var3_3 <- readline(prompt = "input temperature for Amsterdam:");
var4 <- readline(prompt = "input temperature for London:");
var4_2 <- readline(prompt = "input temperature for London:");
var4_3 <- readline(prompt = "input temperature for London:");
var5 <- readline(prompt = "input temperature for Paris:");
var5_2 <- readline(prompt = "input temperature for Paris:");
var5_3 <- readline(prompt = "input temperature for Paris:");
var6 <- readline(prompt = "input temperature for Berlin:");
var6_2 <- readline(prompt = "input temperature for Berlin:");
var6_3 <- readline(prompt = "input temperature for Berlin:");
var7 <- readline(prompt = "input temperature for Madrid:");
var7_2 <- readline(prompt = "input temperature for Madrid:");
var7_3 <- readline(prompt = "input temperature for Madrid:");
var8 <- readline(prompt = "input temperature for Miami:");
var8_2 <- readline(prompt = "input temperature for Miami:");
var8_3 <- readline(prompt = "input temperature for Miami:");
var9 <- readline(prompt = "input temperature for New York:");
var9_2 <- readline(prompt = "input temperature for New York:");
var9_3 <- readline(prompt = "input temperature for New York:");
var10 <- readline(prompt = "input temperature for Los Angeles:");
var10_2 <- readline(prompt = "input temperature for Los Angeles:");
var10_3 <- readline(prompt = "input temperature for Los Angeles:");
```

```
# making it to numeric so it can work with numbers.
```

```
My_Stockholm <-c(var1,var1_2,var1_3)
```

```
a <- as.numeric(My_Stockholm)
```

```
My_Oslo <-c(var2,var2_2,var2_3)
```

```
b <- as.numeric(My_Oslo)
```

```
My_Amsterdam <-c(var3,var3_2,var3_3)
```

```
c <- as.numeric(My_Amsterdam)
```

```
My_London <-c(var4,var4_2,var4_3)
```

```
d <- as.numeric(My_London)
```

```
My_Paris <-c(var5,var5_2,var5_3)
```

```
e <- as.numeric(My_Paris)
```

```
My_Berlin <-c(var6,var6_2,var6_3)
```

```
f <- as.numeric(My_Berlin)
```

```
My_Madrid <-c(var7,var7_2,var7_3)
```

```
g <- as.numeric(My_Madrid)
```

```
My_Miami <-c(var8,var8_2,var8_3)
```

```
h <- as.numeric(My_Miami)
```

```
My_New_York <-c(var9,var9_2,var9_3)
```

```
j <- as.numeric(My_New_York)
```

```
My_Los_Angeles <-c(var10,var10_2,var10_3)
```

```
k <- as.numeric(My_Los_Angeles)
```

#Part A) By using manual calculations did this only with Stockholm

#if else if else statement to see which of them 3 is minimum the prints if the statement filed

```
if (var1 < var1_2 && var1 < var1_3) {  
  cat(var1, " Minimum Temperature for Stockholm\n")  
} else if (var1_2 < var1 && var1_2 < var1_3) {  
  cat(var1_2, " Minimum Temperature for Stockholm\n")  
} else {  
  cat(var1_3, " Minimum Temperature for Stockholm\n")  
}
```

#if else if else Maximum to see which of them 3 is minimum the prints if the statement filed

```
if (var1 > var1_2 && var1 > var1_3) {  
  cat(var1, " Maximum Temperature for Stockholm\n")  
} else if (var1_2 > var1 && var1_2 > var1_3) {  
  cat(var1_2, " Maximum Temperature for Stockholm\n")  
} else {  
  cat(var1_3, " Maximum Temperature for Stockholm\n")  
}
```

#Average

b <- 3

avg = a/b

print(paste("average Temperature for Stockholm\n", avg))

Part B) By using the in-built functions in R

print(paste("Minimum Temperature for Stockholm:", min(a)))

print(paste("Maximum Temperature for Stockholm:", max(a)))

print(paste("average Temperature for Stockholm:", mean(a)))

```
print(paste("Minimum Temperature for Oslo:", min(b)))  
print(paste("Maximum Temperature for Oslo:", max(b)))  
print(paste("average Temperature for Oslo:", mean(b)))
```

```
print(paste("Minimum Temperature for Amsterdam:", min(c)))  
print(paste("Maximum Temperature for Amsterdam:", max(c)))  
print(paste("average Temperature for Amsterdam:", mean(c)))
```

```
print(paste("Minimum Temperature for London:", min(d)))  
print(paste("Maximum Temperature for London:", max(d)))  
print(paste("average Temperature for London:", mean(d)))
```

```
print(paste("Minimum Temperature for Paris:", min(e)))  
print(paste("Maximum Temperature for Paris:", max(e)))  
print(paste("average Temperature for Paris:", mean(e)))
```

```
print(paste("Minimum Temperature for Berlin:", min(f)))  
print(paste("Maximum Temperature for Berlin:", max(f)))  
print(paste("average Temperature for Berlin:", mean(f)))
```

```
print(paste("Minimum Temperature for Madrid:", min(g)))  
print(paste("Maximum Temperature for Madrid:", max(g)))  
print(paste("average Temperature for Madrid:", mean(g)))
```

```
print(paste("Minimum Temperature for Miami:", min(h)))  
print(paste("Maximum Temperature for Miami:", max(h)))  
print(paste("average Temperature for Miami:", mean(h)))
```

```
print(paste("Minimum Temperature for New York:", min(j)))  
print(paste("Maximum Temperature for New York:", max(j)))  
print(paste("average Temperature for New York:", mean(j)))
```

```
print(paste("Minimum Temperature for Los Angeles:", min(k)))  
print(paste("Maximum Temperature for Los Angeles:", max(k)))  
print(paste("average Temperature for Los Angeles:", mean(k)))
```

Task 2) Data Pre-Processing SCRIPT

```
# Reading the file i want to Pre-process
```

```
Dataset = read.csv("kidney_disease.csv", stringsAsFactors = FALSE)
```

```
View(Dataset)
```

```
# Fixing proper formatting
```

```
Dataset$pcv <- as.numeric(Dataset$pcv) #changing Charter to integer/numbers
```

```
Dataset$wc <- as.numeric(Dataset$wc) #changing Charter to integer/numbers
```

```
Dataset$rc <- as.numeric(Dataset$rc) #changing Charter to integer/numbers
```

```
# Corrupted data / missing data for integer/numbers
```

```
mean_age<- as.integer(mean(Dataset$age, na.rm = TRUE)) #now replacing missing data with average  
of the column
```

```
Dataset$age[is.na(Dataset$age)] = mean_age #If NA accouter set it to average.
```

```
mean_bp <- as.integer(mean(Dataset$bp, na.rm = TRUE))
```

```
Dataset$bp[is.na(Dataset$bp)] = mean_bp
```



```
mean_sg <- as.integer(mean(Dataset$sg, na.rm = TRUE))  
Dataset$sg[is.na(Dataset$sg)] = mean_sg
```

```
mean_al <- as.integer(mean(Dataset$al, na.rm = TRUE))  
Dataset$al[is.na(Dataset$al)] = mean_al
```

```
mean_su <- as.integer(mean(Dataset$su, na.rm = TRUE))  
Dataset$su[is.na(Dataset$su)] = mean_su
```

```
mean_bgr <- as.integer(mean(Dataset$bgr, na.rm = TRUE))  
Dataset$bgr[is.na(Dataset$bgr)] = mean_bgr
```

```
mean_bu <- as.integer(mean(Dataset$bu, na.rm = TRUE))  
Dataset$bu[is.na(Dataset$bu)] = mean_bu
```

```
mean_sc <- as.integer(mean(Dataset$sc, na.rm = TRUE))  
Dataset$sc[is.na(Dataset$sc)] = mean_sc
```

```
mean_sod <- as.integer(mean(Dataset$sod, na.rm = TRUE))  
Dataset$sod[is.na(Dataset$sod)] = mean_sod
```

```
mean_pot <- as.integer(mean(Dataset$pot, na.rm = TRUE))  
Dataset$pot[is.na(Dataset$pot)] = mean_pot
```

```
mean_hemo <- as.integer(mean(Dataset$hemo, na.rm = TRUE))  
Dataset$hemo[is.na(Dataset$hemo)] = mean_hemo
```

```
mean_pcv <- as.integer(mean(Dataset$pcv, na.rm = TRUE))  
Dataset$pcv[is.na(Dataset$pcv)] = mean_pcv
```

```
mean_wc <- as.integer(mean(Dataset$wc, na.rm = TRUE))
```

```
Dataset$wc[is.na(Dataset$wc)] = mean_wc
```

```
mean_rc <- as.integer(mean(Dataset$rc, na.rm = TRUE))
```

```
Dataset$rc[is.na(Dataset$rc)] = mean_rc
```

Changing (yes/normal/present/good/ckd and no/abnormal/notpresent/poor/notckd) to 1 and 0 so it's easier for machine to read.

```
Dataset$rbc = factor(Dataset$rbc, levels = c("normal", "abnormal"), labels = c(1, 0)) #changing yes or no to 1 and 0 so it's easier for machine to read.
```

```
Dataset$rbc[is.na(Dataset$rbc)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$pc = factor(Dataset$pc, levels = c("normal", "abnormal"), labels = c(1, 0))
```

```
Dataset$pc[is.na(Dataset$pc)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$pcc = factor(Dataset$pcc, levels = c("present", "notpresent"), labels = c(1, 0))
```

```
Dataset$pcc[is.na(Dataset$pcc)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$ba = factor(Dataset$ba, levels = c("present", "notpresent"), labels = c(1, 0))
```

```
Dataset$ba[is.na(Dataset$ba)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$htn = factor(Dataset$htn, levels = c("yes", "no"), labels = c(1, 0))
```

```
Dataset$htn[is.na(Dataset$htn)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$dm = factor(Dataset$dm, levels = c("yes", "no"), labels = c(1, 0))
```

```
Dataset$dm[is.na(Dataset$dm)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$cad = factor(Dataset$cad, levels = c("yes", "no"), labels = c(1, 0))
```

```
Dataset$cad[is.na(Dataset$cad)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$appet = factor(Dataset$appet, levels = c("good", "poor"), labels = c(1, 0))
```

```
Dataset$appet[is.na(Dataset$appet)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$pe = factor(Dataset$pe, levels = c("yes", "no"), labels = c(1, 0))
```

```
Dataset$pe[is.na(Dataset$pe)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$ane = factor(Dataset$ane, levels = c("yes", "no"), labels = c(1, 0))
```

```
Dataset$ane[is.na(Dataset$ane)] <- 0 # has NA or NULL so fixing to 0
```

```
Dataset$classification = factor(Dataset$classification, levels = c("ckd", "notckd"), labels = c(1, 0))
```

```
Dataset$classification[is.na(Dataset$classification)] <- 0
```

```
str(Dataset) #shows structure on this data.
```

```
head(Dataset, 30) #shows me 30 max from the 400 objects.
```

How it looks more machine reading friendly

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Task 2 and 3.R Dataset

	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	0	1	0	0	121	36.0	1.20	137.0	4.0	15.4	44	7800	5.2	1	1	0	1	0	0	1
2	1	7	50	1.020	4	0	0	1	0	0	148	18.0	0.80	137.0	4.0	11.3	38	6000	4.0	0	0	0	1	0	0	1
3	2	62	80	1.010	2	3	1	1	0	0	423	53.0	1.80	137.0	4.0	9.6	31	7500	4.0	0	1	0	0	0	1	1
4	3	48	70	1.005	4	0	1	0	1	0	117	56.0	3.80	111.0	2.5	11.2	32	6700	3.9	1	0	0	0	1	1	1
5	4	51	80	1.010	2	0	1	1	0	0	106	26.0	1.40	137.0	4.0	11.6	35	7300	4.6	0	0	0	1	0	0	1
6	5	60	90	1.015	3	0	0	0	0	0	74	25.0	1.10	142.0	3.2	12.2	39	7800	4.4	1	1	0	1	1	0	1
7	6	68	70	1.010	0	0	0	1	0	0	100	54.0	24.00	104.0	4.0	12.4	36	8406	4.0	0	0	0	1	0	0	1
8	7	24	76	1.015	2	4	1	0	0	0	410	31.0	1.10	137.0	4.0	12.4	44	6900	5.0	0	1	0	1	1	0	1
9	8	52	100	1.015	3	0	1	0	1	0	138	60.0	1.90	137.0	4.0	10.8	33	9600	4.0	1	1	0	1	0	1	1
10	9	53	90	1.020	2	0	0	0	1	0	70	107.0	7.20	114.0	3.7	9.5	29	12100	3.7	1	1	0	0	0	1	1
11	10	50	60	1.010	2	4	0	0	1	0	490	55.0	4.00	137.0	4.0	9.4	28	8406	4.0	1	1	0	1	0	1	1
12	11	63	70	1.010	3	0	0	0	1	0	380	60.0	2.70	131.0	4.2	10.8	32	4500	3.8	1	1	0	0	1	0	1
13	12	68	70	1.015	3	1	0	1	1	0	208	72.0	2.10	138.0	5.8	9.7	28	12200	3.4	1	1	1	0	1	0	1
14	13	68	70	1.000	1	0	0	0	0	0	98	86.0	4.60	135.0	3.4	9.8	38	8406	4.0	1	1	1	0	1	0	1
15	14	68	80	1.010	3	2	1	0	1	1	157	90.0	4.10	130.0	6.4	5.6	16	11000	2.6	1	1	0	1	0	1	1
16	15	40	80	1.015	3	0	0	1	0	0	76	162.0	9.60	141.0	4.9	7.6	24	3800	2.8	1	0	0	1	0	1	1
17	16	47	70	1.015	2	0	0	1	0	0	99	46.0	2.20	138.0	4.1	12.6	38	8406	4.0	0	0	0	1	0	0	1
18	17	47	80	1.000	1	0	0	0	0	0	114	87.0	5.20	139.0	3.7	12.1	38	8406	4.0	1	0	0	0	0	0	1
19	18	60	100	1.025	0	3	0	1	0	0	263	27.0	1.30	135.0	4.3	12.7	37	11400	4.3	1	1	1	1	0	0	1
20	19	62	60	1.015	1	0	0	0	1	0	100	31.0	1.60	137.0	4.0	10.3	30	5300	3.7	1	0	1	1	0	0	1
21	20	61	80	1.015	2	0	0	0	0	0	173	148.0	3.90	135.0	5.2	7.7	24	9200	3.2	1	1	1	0	1	1	1
22	21	60	90	1.000	1	0	0	0	0	0	148	180.0	76.00	4.5	4.0	10.9	32	6200	3.6	1	1	1	1	0	0	1
23	22	48	80	1.025	4	0	1	0	0	0	95	163.0	7.70	136.0	3.8	9.8	32	6900	3.4	1	0	0	1	0	1	1
24	23	21	70	1.010	0	0	0	1	0	0	148	57.0	3.00	137.0	4.0	12.0	38	8406	4.0	0	0	0	0	0	1	1
25	24	42	100	1.015	4	0	1	0	0	1	148	50.0	1.40	129.0	4.0	11.1	39	8300	4.6	1	0	0	0	0	0	1
26	25	61	60	1.025	0	0	0	1	0	0	108	75.0	1.90	141.0	5.2	9.9	29	8400	3.7	1	1	0	1	0	1	1
27	26	75	80	1.015	0	0	0	1	0	0	156	45.0	2.40	140.0	3.4	11.6	35	10300	4.0	1	1	0	0	0	0	1
28	27	69	70	1.010	3	4	1	0	0	0	264	87.0	2.70	130.0	4.0	12.5	37	9600	4.1	1	1	1	1	1	0	1
29	28	75	70	1.000	1	3	0	0	0	0	123	31.0	1.40	137.0	4.0	12.0	38	8406	4.0	0	1	0	1	0	0	1
30	29	68	70	1.005	1	0	0	0	1	0	148	28.0	1.40	137.0	4.0	12.9	38	8406	4.0	0	0	1	1	0	0	1
31	30	51	70	1.000	1	0	0	0	0	0	93	155.0	7.30	132.0	4.9	12.0	38	8406	4.0	1	0	0	1	0	0	1
32	31	73	90	1.015	3	0	0	0	1	0	107	33.0	1.50	141.0	4.6	10.1	30	7800	4.0	0	0	0	0	0	0	1
33	32	61	90	1.010	1	1	0	1	0	0	159	39.0	1.50	133.0	4.9	11.3	34	9600	4.0	1	1	0	0	0	0	1
34	33	60	100	1.020	2	0	0	0	0	0	140	55.0	2.50	137.0	4.0	10.1	29	8406	4.0	1	0	0	0	0	0	1
35	34	70	70	1.010	1	0	1	0	1	1	171	153.0	5.20	137.0	4.0	12.0	38	8406	4.0	0	1	0	0	0	0	1
36	35	65	90	1.020	2	1	0	1	0	0	270	39.0	2.00	137.0	4.0	12.0	36	9800	4.9	1	1	0	0	0	1	1
37	36	76	70	1.015	1	0	1	1	0	0	92	29.0	1.80	133.0	3.9	10.3	32	8406	4.0	1	0	0	1	0	0	1
38	37	72	80	1.000	1	0	0	0	0	0	137	65.0	3.40	141.0	4.7	9.7	28	6900	2.5	1	1	0	0	0	1	0
39	38	69	80	1.020	3	0	0	1	0	0	148	103.0	4.10	132.0	5.9	12.5	38	8406	4.0	1	0	0	1	0	0	1
40	39	82	80	1.010	2	2	1	0	0	0	140	70.0	3.40	136.0	4.2	13.0	40	9800	4.2	1	1	0	1	0	0	1
41	40	46	90	1.010	2	0	1	0	0	0	99	80.0	2.10	137.0	4.0	11.1	32	9100	4.1	1	0	0	1	0	0	1
42	41	45	70	1.010	0	0	0	1	0	0	148	20.0	0.70	137.0	4.0	12.0	38	8406	4.0	0	0	0	1	1	0	1
43	42	47	100	1.010	0	0	0	1	0	0	204	29.0	1.00	139.0	4.2	9.7	33	9200	4.5	1	0	0	1	0	1	1
44	43	35	80	1.010	1	0	0	0	0	0	79	202.0	10.80	134.0	3.4	7.9	24	7900	3.1	0	1	0	1	0	0	1
45	44	54	80	1.010	3	0	0	0	0	0	207	77.0	6.30	134.0	4.8	9.7	28	8406	4.0	1	1	0	0	1	0	1
46	45	54	80	1.020	3	0	0	0	0	0	208	89.0	5.90	130.0	4.9	9.3	38	8406	4.0	1	1	0	0	1	0	1
47	46	48	70	1.015	0	0	0	1	0	0	124	24.0	1.20	142.0	4.2	12.4	37	6400	4.7	0	1	0	1	0	0	1
48	47	11	80	1.010	3	0	0	1	0	0	148	17.0	0.80	137.0	4.0	15.0	45	8600	4.0	0	0	0	1	0	0	1
49	48	73	70	1.005	0	0	1	1	0	0	70	32.0	0.90	125.0	4.0	10.0	29	18900	3.5	1	1	0	1	1	0	1

Showing 1 to 50 of 400 entries, 26 total columns

Task 3) Data Visualization: Im using same file as question 2 dataset

```
library(ggplot2) #launching ggplot
```

```
#Using GGPlot2
```

```
g <- ggplot(Dataset, aes(x=sod, y=bp)) + geom_point(aes(col=age,size=su)) +  
coord_cartesian(xlim=c(0,163), ylim=c(0, 180)) + labs(title="sodium(sod) Vs Blood pressure(bp)",  
subtitle="Chronic Kidney Disease Data Set", y="Blood pressure", x="Sodium", col="Age", size="Sugar  
1-5", caption="Dr.P.Soundarapandian.M.D.,D.M")
```

```
plot(g)
```

So sodium on the x axel and blood pressure on the y axel, also on the col is age with color and size as sugar from 1 to 5. On the x axel I went to 163 because it was the highest number on that topic and 180 on blood pressure.

