Assignment R programming

Ekaterina Zhiganova

Data Scientist, EC Utbildning, Solna, 2022

Basic algorithmic thinking in R

Users data:

Malmö/ Malmo
14,14,12,14,13,14,12,11,12,13,12,11,10,10,11,13,12,13,14,12,11,10,10,9,9,10,8,7,9,6

Gothenburg
13,13,11,12,12,12,10,12,10,11,11,10,8,9,10,11,12,12,12,11,10,10,11,10,8,10,7,6,7,6

Stockholm 12,11,10,10,11,11,11,10,10,10,11,11,7,7,8,10,11,10,10,11,10,10,8,10,7,9,8,7,7,6

Sundsvall 11,11,9,9,10,8,9,8,7,8,10,10,8,6,7,8,6,5,7,7,7,8,7,6,5,5,5,6,7,5

Östersund/Ostersund 10,10,9,8,11,7,8,8,9,9,10,9,8,7,7,8,7,7,7,7,7,6,7,6,6,6,5,6,7,6

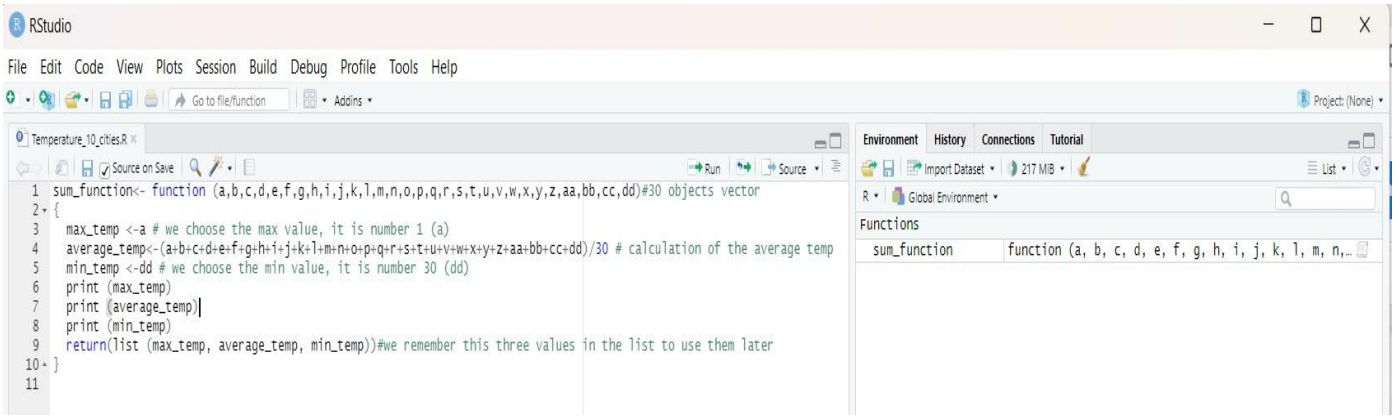
Luleå/Lulea 10,8,9,10,10,7,8,7,9,9,8,9,8,7,6,8,7,6,7,6,7,6,7,6,5,6,5,6,4,4

Umeå/Umea 9,8,7,8,9,7,8,7,6,5,7,6,7,5,6,7,5,6,7,6,5,6,5,6,4,6,5,3,4,3

Kiruna 7,5,7,6,7,4,5,6,5,4,6,5,5,4,4,6,5,6,4,5,3,4,4,3,2,3,4,2,2,2

My work:

Firstly I create a program to manual calculation temperature dependence of the data:



Here I use the function with 30 values and set maximum temperature as the first value in the vector, minimum value as the last value in the vector(it can be another member, I just did it manually) and the average temperature as a sum of all values decrease by their number.

I also return those three values in the list, to use them later.

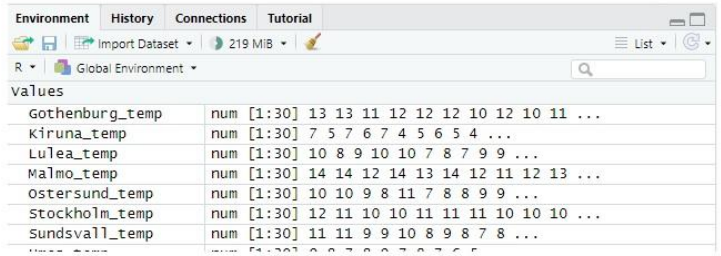
After that I did the data input, as for Malmö:

```
Console  Terminal  Background Jobs
R 4.2.1 · ~/
> Malmo_temp <-c(14,14,12,14,13,14,12,11,12,13,12,11,10,10,11,13,12,13,14,12,11,10,10,9,9,10,8,7,9,6)
>
```

We can see that Malmo_temp now looks like a vector with 30 values in it.

I did the same with 9 other cities:

```
Console  Terminal  Background Jobs
R 4.2.1 · ~/
> Malmo_temp <-c(14,14,12,14,13,14,12,11,12,13,12,11,10,10,11,13,12,13,14,12,11,10,10,9,9,10,8,7,9,6)
> Gothenburg_temp <- c(13,13,11,12,12,12,10,12,10,11,11,10,8,9,10,11,12,12,12,11,10,10,11,10,8,10,7,6,7,6)
> Stockholm_temp <- c(12,11,10,10,11,11,11,10,10,10,11,11,7,7,8,10,11,10,10,11,10,10,8,10,7,9,8,7,7,6)
> Sundsvall_temp <- c(11,11,9,9,10,8,9,8,7,8,10,10,8,6,7,8,6,5,7,7,7,8,7,6,5,5,5,6,7,5)
> Ostersund_temp <- c(10,10,9,8,11,7,8,8,9,9,10,9,8,7,7,8,7,7,7,7,7,6,7,6,6,6,5,6,7,6)
> Lulea_temp <-c(10,8,9,10,10,7,8,7,9,9,8,9,8,7,6,8,7,6,7,6,7,6,7,6,5,6,5,6,4,4)
> Umea_temp <-c(9,8,7,8,9,7,8,7,6,5,7,6,7,5,6,7,5,6,7,6,5,6,5,6,4,6,5,3,4,3)
> Kiruna_temp <-c(7,5,7,6,7,4,5,6,5,4,6,5,5,4,4,6,5,6,4,5,3,4,4,3,2,3,4,2,2,2)
>
```



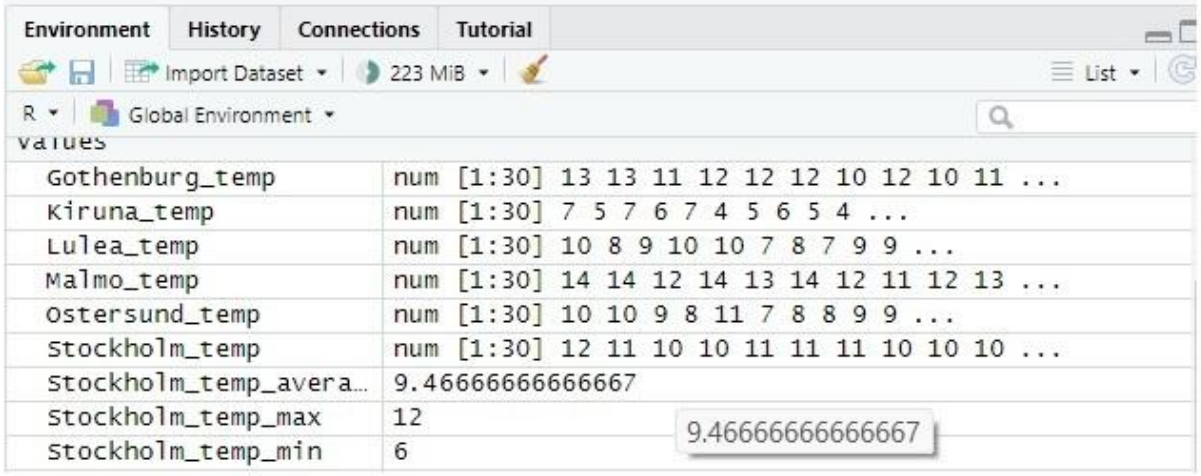
After that, I did the manual calculation of the minimum, maximum and average temperature for the Stockholm city by using sum_function I wrote in the beginning:

```
> Manual_Stockholm_temp <-sum_function(12,11,10,10,11,11,11,10,10,10,11,11,7,7,8,10,11,10,10,11,10,10,8,10,7,9,8,7,7,6)
[1] 12
[1] 9.466667
[1] 6
>
```

As we see, in Environment is a list with all 3 values.

And In-Build R calculation for the Stockholm with min(), max() and mean() functions:

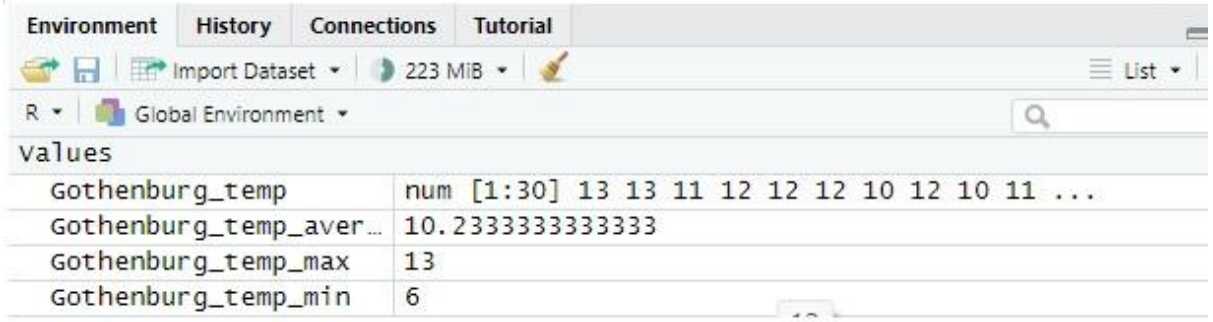
```
> Stockholm_temp_min<-min(Stockholm_temp)
> Stockholm_temp_max <-max(Stockholm_temp)
> Stockholm_temp_averaga<-mean(Stockholm_temp)
>
```



Next, manual and In-Build calculations for the Gothenborg

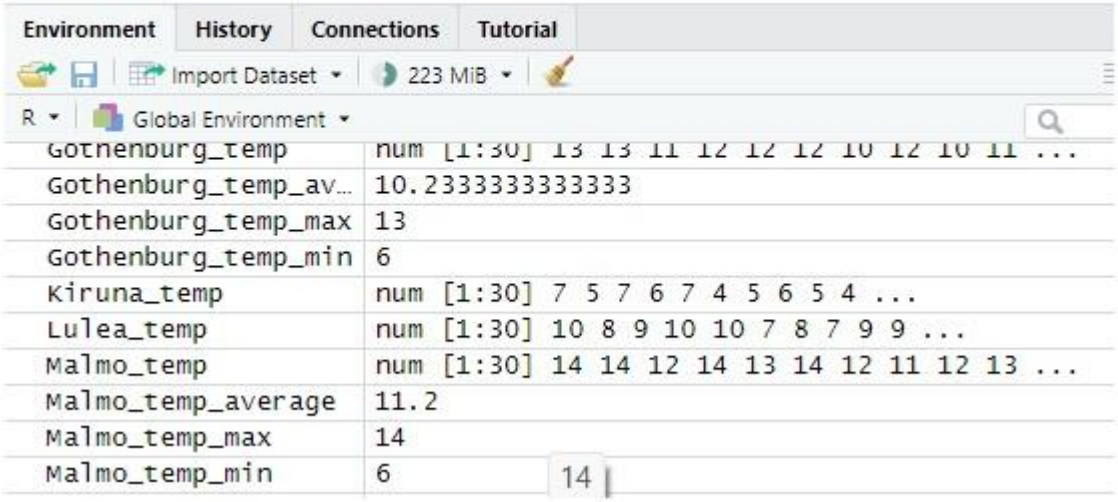
```
> Manual_Gothenburg_temp <-sum_function(13,13,11,12,12,12,10,12,10,11,11,10,8,9,10,11,12,12,12,11,10,10,11,10,8,10,7,6,7,6)
[1] 13
[1] 10.23333
[1] 6
>
```


```
> Gothenburg_temp_min<-min(Gothenburg_temp)
> Gothenburg_temp_max <-max(Gothenburg_temp)
> Gothenburg_temp_average<-mean(Gothenburg_temp)
>
```

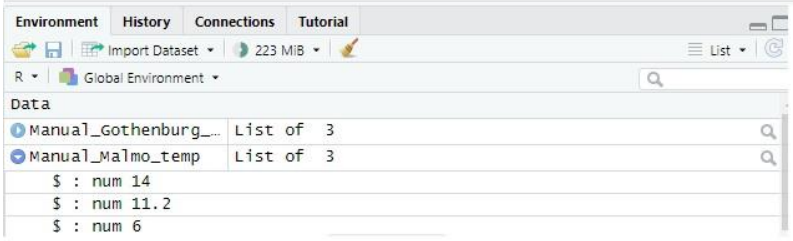


For the Malmö

```
> Malmo_temp_min<-min(Malmo_temp)
> Malmo_temp_max<-max(Malmo_temp)
> Malmo_temp_average<-mean(Malmo_temp)
>
```

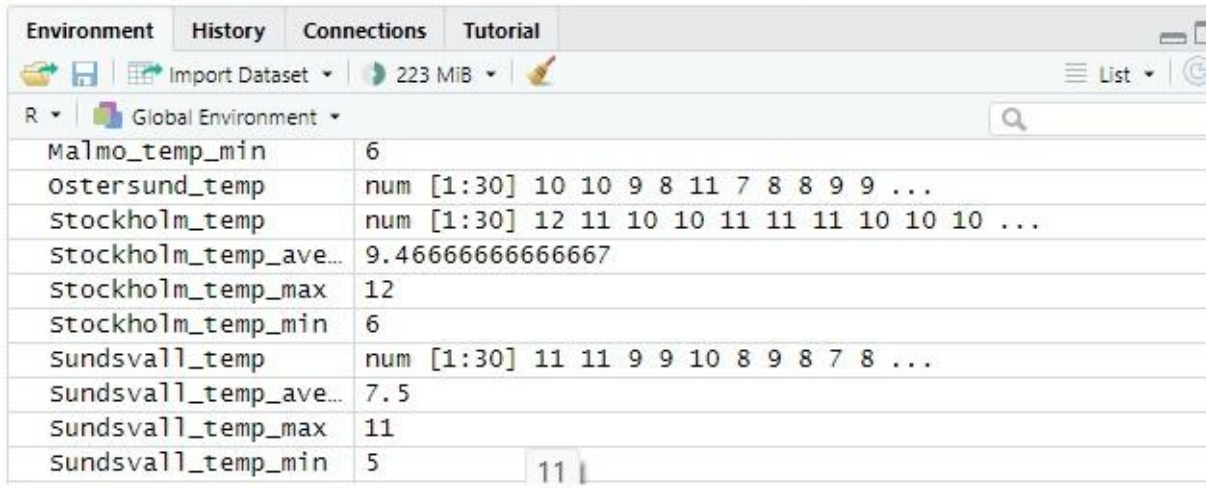



```
> Manual_Malmo_temp <-sum_function(14,14,12,14,13,14,12,11,12,13,12,11,10,10,11,13,12,13,14,12,11,10,10,9,9,10,8,7,9,6)
[1] 14
[1] 11.2
[1] 6
>
```



For the Sundsvall

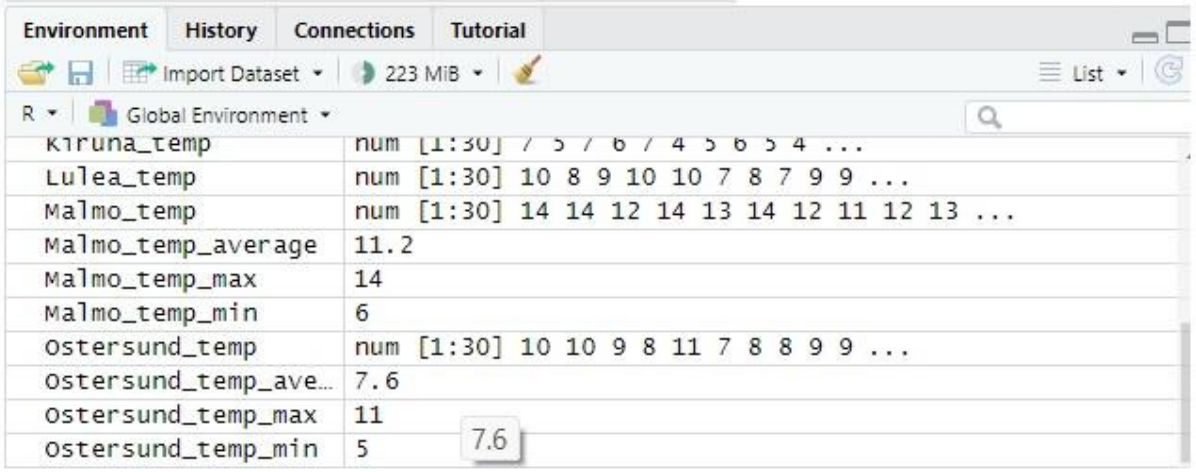
```
> Sundsvall_temp_min<-min(Sundsvall_temp)
> Sundsvall_temp_max <-max(Sundsvall_temp)
> Sundsvall_temp_average <-mean(Sundsvall_temp)
>
```



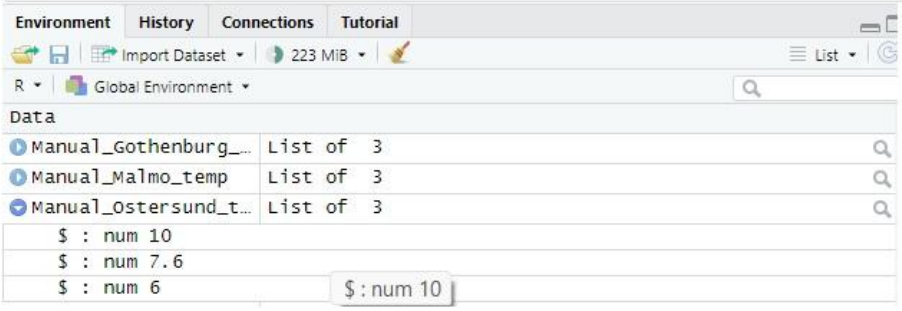
```
> Manual_Sundsvall_temp <-sum_function(11,11,9,9,10,8,9,8,7,8,10,10,8,6,7,8,6,5,7,7,7,8,7,6,5,5,5,6,7,5)
[1] 11
[1] 7.5
[1] 5
>
```

For the Östersund

```
> Ostersund_temp_min <-min(Ostersund_temp)
> Ostersund_temp_max<-max(Ostersund_temp)
> Ostersund_temp_average<-mean(Ostersund_temp)
>
```

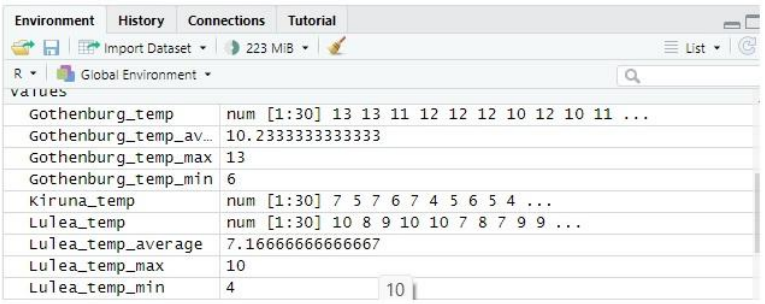



```
> Manual_Ostersund_temp <-sum_function(10,10,9,8,11,7,8,8,9,9,10,9,8,7,7,8,7,7,7,7,7,6,7,6,6,6,5,6,7,6)
[1] 10
[1] 7.6
[1] 6
>
```

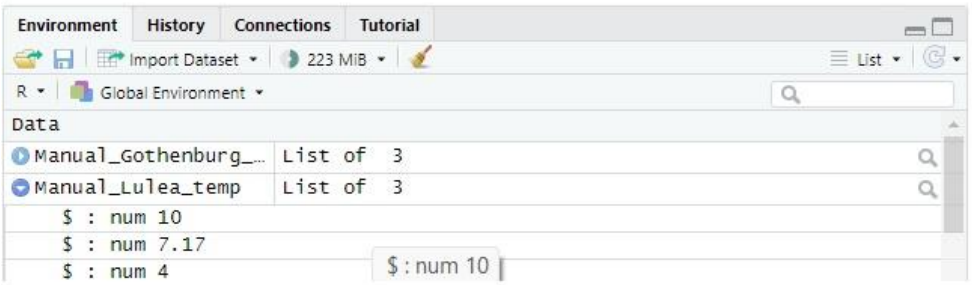


For the Luleå

```
> Lulea_temp_min <-min(Lulea_temp)
> Lulea_temp_max <-max(Lulea_temp)
> Lulea_temp_average <-mean(Lulea_temp)
>
```

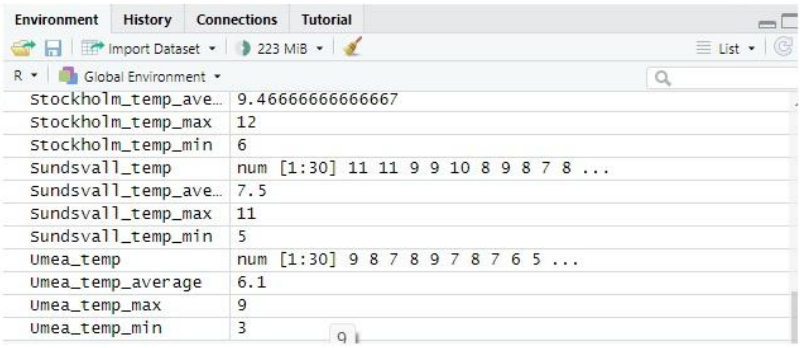



```
> Manual_Lulea_temp <-sum_function(10,8,9,10,10,7,8,7,9,9,8,9,8,7,6,8,7,6,7,6,7,6,7,6,5,6,5,6,4,4)
[1] 10
[1] 7.166667
[1] 4
>
```

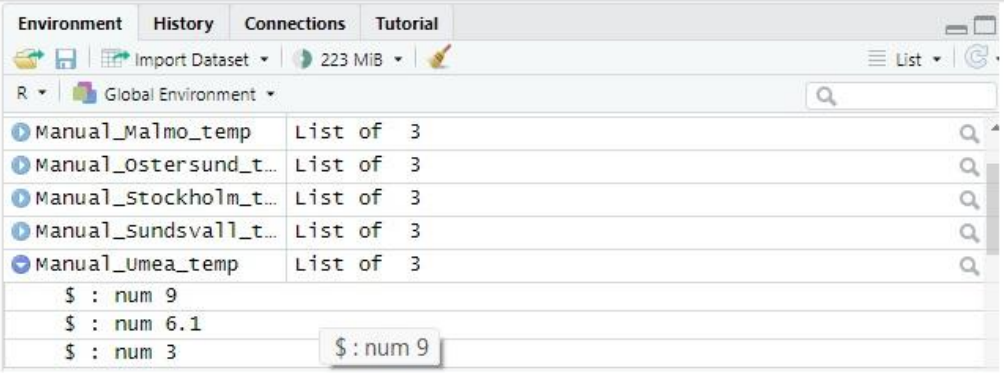


For the Umeå

```
> Umea_temp_min<-min(Umea_temp)
> Umea_temp_max<-max(Umea_temp)
> Umea_temp_average<-mean(Umea_temp)
>
```

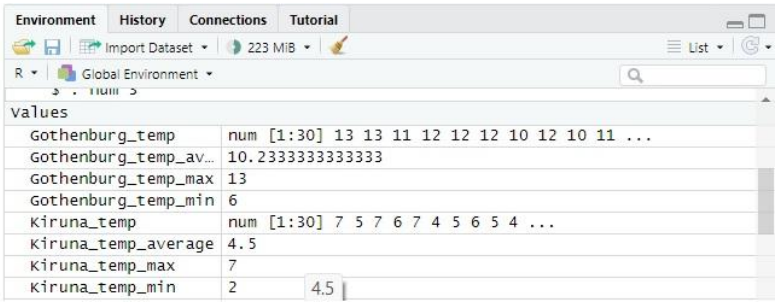



```
> Manual_Umea_temp <-sum_function(9,8,7,8,9,7,8,7,6,5,7,6,7,5,6,7,5,6,7,6,5,6,5,6,4,6,5,3,4,3)
[1] 9
[1] 6.1
[1] 3
>
```

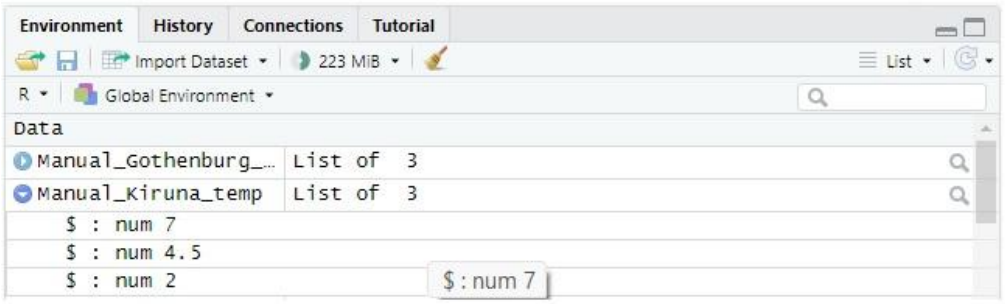


For the Kiruna

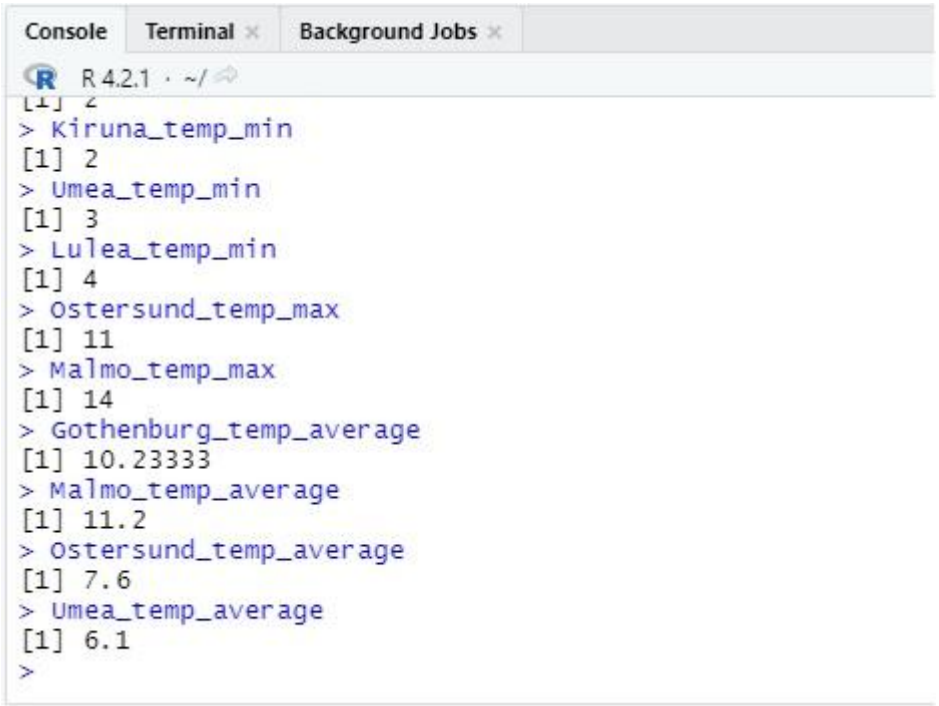
```
> Kiruna_temp_min<-min(Kiruna_temp)
> Kiruna_temp_max <-max(Kiruna_temp)
> Kiruna_temp_average<-mean(Kiruna_temp)
>
```




```
> Manual_Kiruna_temp<-sum_function(7,5,7,6,7,4,5,6,5,4,6,5,5,4,4,6,5,6,4,5,3,4,4,3,2,3,4,2,2,2)
[1] 7
[1] 4.5
[1] 2
>
```



We can always display our values on the console as:



Data pre-processing

I used the kidney_disease.csv file. Here are the 400 objects of 26 variables (id, age and characteristics of health).

Firstly I download a file and View it with the View() function.

I can see all the data, there are few missing numbers in the “age” column. I insert the average age in the empty places.

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

Then I see in the “bp” column some missing numbers, but I can’t insert the average namber, because “bp” is a decrease of 10 numbers. I calculate an average number with mean() function - it’s 76, and increase the 80 in the missing places.

```
> mean_bp<-as.integer(mean(Dataset$bp, na.rm = TRUE))
> Dataset$bp[is.na(Dataset$bp)] = 80
>
```

In the “sg” column I calculate an average, it’s 1.0174. I change empty places to 1.015.

```
> mean_sg<-as.numeric(mean(Dataset$sg, na.rm = TRUE))
> Dataset$sg[is.na(Dataset$sg)] = 1.015
```

“al” and “su” are numeric between 0 and 5. I insert an average in the empty spaces.

```
> 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
>
```

“pc” and “rbc” I change “normal” and “abnormal” to 1 and 0. Empty spaces will be 0.

```
> Dataset$rbc =factor(Dataset$rbc, levels = c('normal', 'abnormal'), labels=c(1,0))
> Dataset$rbc[is.na(Dataset$rbc)] <- 3
Warning message:
In `[<-.factor`(`*tmp*`, is.na(Dataset$rbc), value = c(NA, NA, 1L,  :
  invalid factor level, NA generated
> Dataset$rbc[is.na(Dataset$rbc)] <- 0
```

```
> Dataset$pc =factor(Dataset$pc, levels = c('normal', 'abnormal'), labels=c(1,0))
> Dataset$pc[is.na(Dataset$pc)] <- 0
>
```

“pcc” and “ba” I change “present” and “notpresent” to 1 and 0. Empty spaces will be 0.

```
> Dataset$pcc =factor(Dataset$pcc, levels = c('notpresent', 'present'), labels=c(0,1))
> Dataset$pcc[is.na(Dataset$pcc)] <-0
> Dataset$ba =factor(Dataset$ba, levels = c('notpresent', 'present'), labels=c(0,1))
> Dataset$ba[is.na(Dataset$ba)] <-0
>
```

“bgr” there are few missing numbers, I insert the average “bgr” in the empty places.

“bu” I calculate an average, it’s 57 and insert 57.0 as a right image of the number.

“sc” I calculate an average, it’s 3.07 and insert 3.1.

“sod” I calculate an average, it’s 137 and insert 137.0 as a right image of the number.

```
> 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)] = 57.0
> mean_sc<-as.numeric(mean(Dataset$sc, na.rm = TRUE))
> Dataset$sc[is.na(Dataset$sc)] = 3.1
> mean_sod<-as.integer(mean(Dataset$sod, na.rm = TRUE))
> Dataset$sod[is.na(Dataset$sod)] = 137.0
>
```

“pot” I calculate an average, it’s 4.62 and insert 4.6.

“hemo” I calculate an average, it’s 12.52 and insert 12.5.

```
> mean_pot<-as.numeric(mean(Dataset$pot, na.rm = TRUE))
> Dataset$pot[is.na(Dataset$pot)] = 4.6
> mean_hemo<-as.numeric(mean(Dataset$hemo, na.rm = TRUE))
> Dataset$hemo[is.na(Dataset$hemo)] = 12.5
>
```

“pcv” has characteristics, I change it to integer, calculate and insert an average number.

```
> mean_pcv<-as.integer(mean(Dataset$pcv, na.rm = TRUE))
Warning message:
In mean.default(Dataset$pcv, na.rm = TRUE) :
  argument is not numeric or logical: returning NA
> Dataset$pcv<-as.integer(Dataset$pcv)
Warning message:
NAs introduced by coercion
> mean_pcv<-as.integer(mean(Dataset$pcv, na.rm = TRUE))
> Dataset$pcv[is.na(Dataset$pcv)] = mean_pcv
>
```

“wc” is also not integer, I change it to integer, calculate an average and insert 8400.

```
> Dataset$wc<-as.integer(Dataset$wc)
Warning message:
NAs introduced by coercion
> mean_wc<-as.integer(mean(Dataset$wc, na.rm = TRUE))
> Dataset$wc[is.na(Dataset$wc)] = 8400
>
```

“rc” is not numeric, I change it to numeric, calculate an average and insert 4.7

```
> mean_rc<-as.numeric(mean(Dataset$rc, na.rm = TRUE))
Warning message:
In mean.default(Dataset$rc, na.rm = TRUE) :
  argument is not numeric or logical: returning NA
> Dataset$rc<-as.numeric(Dataset$rc)
Warning message:
NAs introduced by coercion
> mean_rc<-as.numeric(mean(Dataset$rc, na.rm = TRUE))
```

“htn”, “dm”, “cad”, “pe”, “ane” I change “yes” and “no” to 1 and 0. Empty spaces will be 0.

“appet” I change “good” and “poor” to 1 and 0. Empty spaces will be 0.

“classification” I change “ckd” and “notckd” to 1 and 0. Empty spaces will be 0.

```
> Dataset$htn =factor(Dataset$htn, levels = c('yes', 'no'), labels=c(1,0))
> Dataset$htn[is.na(Dataset$htn)] <-0
> Dataset$dm =factor(Dataset$dm, levels = c('yes', 'no'), labels=c(1,0))
> Dataset$dm[is.na(Dataset$dm)] <-0
> Dataset$cad =factor(Dataset$cad, levels = c('yes', 'no'), labels=c(1,0))
> Dataset$cad[is.na(Dataset$cad)] <-0
> Dataset$pe =factor(Dataset$pe, levels = c('yes', 'no'), labels=c(1,0))
> Dataset$pe[is.na(Dataset$pe)] <-0
> Dataset$ane =factor(Dataset$ane, levels = c('yes', 'no'), labels=c(1,0))
> Dataset$ane[is.na(Dataset$ane)] <-0
> Dataset$appet =factor(Dataset$appet, levels = c('good', 'poor'), labels=c(1,0))
> Dataset$appet[is.na(Dataset$appet)] <-0
> Dataset$classification =factor(Dataset$classification, levels = c('ckd', 'notckd'), labels=c(1,0))
> Dataset$classification[is.na(Dataset$classification)] <-0
>
```

Now it’s clean and ready to be used.

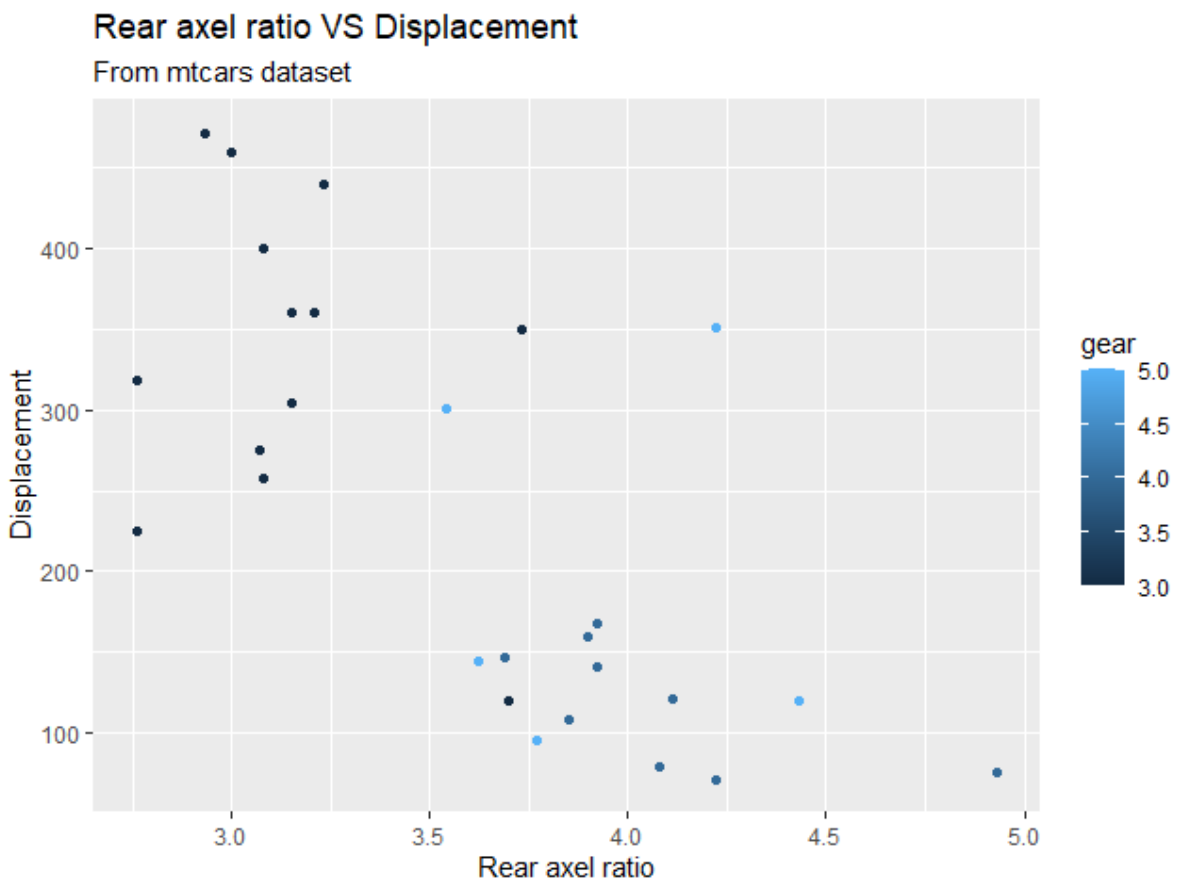
Filter

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm
387	386	46	70	1.025	0	0	1	1	0	0	100	47	0.5	142	3.5	16.4	43	5700	6.5	0	0
388	387	15	80	1.025	0	0	1	1	0	0	93	17	0.9	136	3.9	16.7	50	6200	5.2	0	0
389	388	51	80	1.020	0	0	1	1	0	0	94	15	1.2	144	3.7	15.5	46	9500	6.4	0	0
390	389	41	80	1.025	0	0	1	1	0	0	112	48	0.7	140	5.0	17.0	52	7200	5.8	0	0
391	390	52	80	1.025	0	0	1	1	0	0	99	25	0.8	135	3.7	15.0	52	6300	5.3	0	0
392	391	36	80	1.025	0	0	1	1	0	0	85	16	1.1	142	4.1	15.6	44	5800	6.3	0	0
393	392	57	80	1.020	0	0	1	1	0	0	133	48	1.2	147	4.3	14.8	46	6600	5.5	0	0
394	393	43	60	1.025	0	0	1	1	0	0	117	45	0.7	141	4.4	13.0	54	7400	5.4	0	0
395	394	50	80	1.020	0	0	1	1	0	0	137	46	0.8	139	5.0	14.1	45	9500	4.6	0	0
396	395	55	80	1.020	0	0	1	1	0	0	140	49	0.5	150	4.9	15.7	47	6700	4.9	0	0
397	396	42	70	1.025	0	0	1	1	0	0	75	31	1.2	141	3.5	16.5	54	7800	6.2	0	0
398	397	12	80	1.020	0	0	1	1	0	0	100	26	0.6	137	4.4	15.8	49	6600	5.4	0	0
399	398	17	60	1.025	0	0	1	1	0	0	114	50	1.0	135	4.9	14.2	51	7200	5.9	0	0
400	399	58	80	1.025	0	0	1	1	0	0	131	18	1.1	141	3.5	15.8	53	6800	6.1	0	0

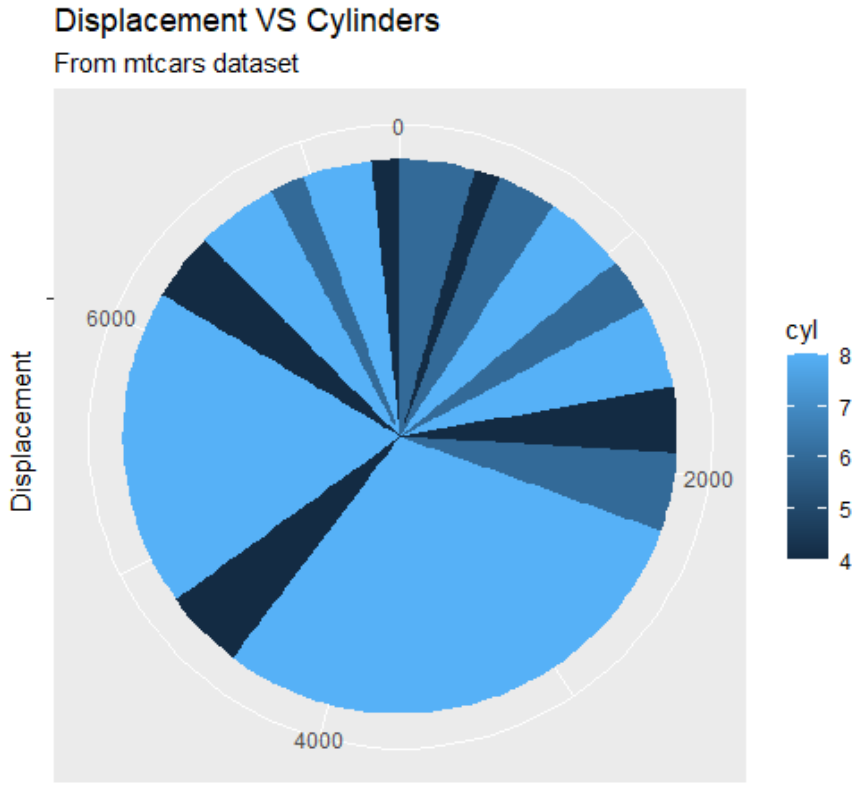
Data Visualization

I used mtcars.csv dataset to visualize some sort of data.

First is a plot, related to rear axle ratio and displacement. Colors indicate a number of gears.



Second is a pie, indicated correlation displacement and a number of cylinders.



Firth is a histogram, indicating a number of cars with different gears.