Descriptive Statistics: the Marital status edition

M P Gururajan, Hina A Gokhale and Dayadeep Monder

Indian Institute of Technology Bombay, Mumbai

In this session, we are going to use a real world dataset. The dataset is called `MaritalStatusAgeWiseIndia.csv` and is stored in `csv` format. The data lists the marital status, age-wise, in India and the different states, for both men and women.

As a first step, and to remind you of some of the commands, we are going to repeat all the plots and analysis we did for the `RKNGT.csv` data set first. Of course, there are differences; some stem from the fact that real world data does not come in a nice package; at times we may have data that we do not need; so, we need to learn how to separate the data we want; some stem from the complexity of the data; the same command might result in a different type of response from R, for example. In addition, complex data also allows for more involved visualisation that is not possible in a very simple data set.

In most cases, it is a good idea to take a look at the data before processing them in R. In my case, I use LibreOffice Calc to look at the data. Immediately, one can see that the data is not in as easy a form to work with as our earlier data set. For example, the data starts from the seventh row. No single line can be considered as a the header – except row 5 which labels most of the data columns; even that line does not label some of the columns of data. How do we deal with this and load and read the data in the form in which we want it? Here is where reading the help file and documentation (and, if needed, some online material or a text book) will help!

Let us first load the data. Note that we want to skip the first four lines, we want the fith line read as the header, and the sixth line to be skipped:

```
unlink("~/.RData")
X <- read.csv("../Data/MaritalStatusAgeWiseIndia.csv",skip=4,
              header=TRUE,blank.lines.skip = TRUE)
str(X)
```

```
## 'data.frame': 2161 obs. of  27 variables:
##  $ X  : Factor w/ 2 levels "","C0402": 1 2 2 2 2 2 2 2 2 2 ...
##  $ X.1: int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ X.2: int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ X.3: Factor w/ 37 levels "","INDIA","State - ANDAMAN & NICOBAR ISLANDS (35)",..: 1 2 2 2 2 2 2 2
##  $ X.4: Factor w/ 4 levels "","Rural","Total",..: 1 3 3 3 3 3 3 3 3 3 ...
##  $ X1 : Factor w/ 21 levels "","0-9","10-14",..: 1 19 2 3 4 5 6 7 8 9 ...
##  $ X2 : int  NA 1210854977 239734904 132709212 120526449 111424222 101413965 88594951 85140684 724381
##  $ X3 : int  NA 623270258 124932540 69418835 63982396 57584693 51344208 44660674 42919381 37545386 ..
##  $ X4 : int  NA 587584719 114802364 63290377 56544053 53839529 50069757 43934277 42221303 34892726 ..
##  $ X5 : int  NA 570833969 239734904 129790438 106188765 56175036 20948980 6480174 2731904 1558004 ...
##  $ X6 : int  NA 322870527 124932540 68312261 60873986 39810362 16551438 5040404 1961245 1053138 ...
##  $ X7 : int  NA 247963442 114802364 61478177 45314779 16364674 4397542 1439770 770659 504866 ...
##  $ X8 : int  NA 579584783 0 2742714 14010633 54417955 78981469 79987358 79349364 67014828 ...
##  $ X9 : int  NA 286507311 0 1032903 2999621 17536278 34374811 39056059 40230399 35626363 ...
##  $ X10: int  NA 293077472 0 1709811 11011012 36881677 44606658 40931299 39118965 31388465 ...
```

```
##  $ X11: int  NA 55538707 0 100477 193733 437226 873244 1439180 2328273 3224938 ...
##  $ X12: int  NA 12277229 0 36830 63249 125774 230746 343170 488916 655785 ...
##  $ X13: int  NA 43261478 0 63647 130484 311452 642498 1096010 1839357 2569153 ...
##  $ X14: int  NA 3535202 0 63478 101818 276865 422608 476673 517039 461425 ...
##  $ X15: int  NA 1162448 0 31232 37223 81864 128577 148419 163218 147730 ...
##  $ X16: int  NA 2372754 0 32246 64595 195001 294031 328254 353821 313695 ...
##  $ X17: int  NA 1362316 0 12105 31500 117140 187664 211566 214104 178917 ...
##  $ X18: int  NA 452743 0 5609 8317 30415 58636 72622 75603 62370 ...
##  $ X19: int  NA 909573 0 6496 23183 86725 129028 138944 138501 116547 ...
##  $ X20: int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ X21: int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ X22: int  NA 0 0 0 0 0 0 0 0 0 ...
```

We see that the sixth row is not skipped and is instead is marked as `NA`. This can cause problems. For example, try

```
mean(X$X10)

## [1] NA
```

The mean turns out to be NA!

Let us try again – this time, let us explicitly ask R to skip the sixth line! To do this, we are going to ue a trick! First, we are going to read all the lines of the file into a list. Then, we prepare a list in which we skip the sixth line. From this new list, we read our data! This might not be the only trick or even the simplest or most elegant trick! But, it does the job for us!!

```
AllLines = readLines("../Data/MaritalStatusAgeWiseIndia.csv")
RemoveSixth = AllLines[-6]
X <- read.csv(textConnection(RemoveSixth),skip=4,
              header=TRUE,blank.lines.skip = TRUE)
str(X)

## 'data.frame': 2160 obs. of  27 variables:
##  $ X  : Factor w/ 1 level "C0402": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X.1: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X.2: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X.3: Factor w/ 36 levels "INDIA","State - ANDAMAN & NICOBAR ISLANDS (35)",..: 1 1 1 1 1 1 1 1 1 1
##  $ X.4: Factor w/ 3 levels "Rural","Total",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ X1 : Factor w/ 20 levels "0-9","10-14",..: 18 1 2 3 4 5 6 7 8 9 ...
##  $ X2 : int  1210854977 239734904 132709212 120526449 111424222 101413965 88594951 85140684 72438112
##  $ X3 : int  623270258 124932540 69418835 63982396 57584693 51344208 44660674 42919381 37545386 32138
##  $ X4 : int  587584719 114802364 63290377 56544053 53839529 50069757 43934277 42221303 34892726 30180
##  $ X5 : int  570833969 239734904 129790438 106188765 56175036 20948980 6480174 2731904 1558004 100220
##  $ X6 : int  322870527 124932540 68312261 60873986 39810362 16551438 5040404 1961245 1053138 639703 .
##  $ X7 : int  247963442 114802364 61478177 45314779 16364674 4397542 1439770 770659 504866 362504 ...
##  $ X8 : int  579584783 0 2742714 14010633 54417955 78981469 79987358 79349364 67014828 56741611 ...
##  $ X9 : int  286507311 0 1032903 2999621 17536278 34374811 39056059 40230399 35626363 30533274 ...
##  $ X10: int  293077472 0 1709811 11011012 36881677 44606658 40931299 39118965 31388465 26208337 ...
##  $ X11: int  55538707 0 100477 193733 437226 873244 1439180 2328273 3224938 4072051 ...
##  $ X12: int  12277229 0 36830 63249 125774 230746 343170 488916 655785 802340 ...
##  $ X13: int  43261478 0 63647 130484 311452 642498 1096010 1839357 2569153 3269711 ...
##  $ X14: int  3535202 0 63478 101818 276865 422608 476673 517039 461425 369587 ...
```

```
##  $ X15: int  1162448 0 31232 37223 81864 128577 148419 163218 147730 117276 ...
##  $ X16: int  2372754 0 32246 64595 195001 294031 328254 353821 313695 252311 ...
##  $ X17: int  1362316 0 12105 31500 117140 187664 211566 214104 178917 132871 ...
##  $ X18: int  452743 0 5609 8317 30415 58636 72622 75603 62370 45521 ...
##  $ X19: int  909573 0 6496 23183 86725 129028 138944 138501 116547 87350 ...
##  $ X20: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X21: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X22: int  0 0 0 0 0 0 0 0 0 0 ...
```

Exploration

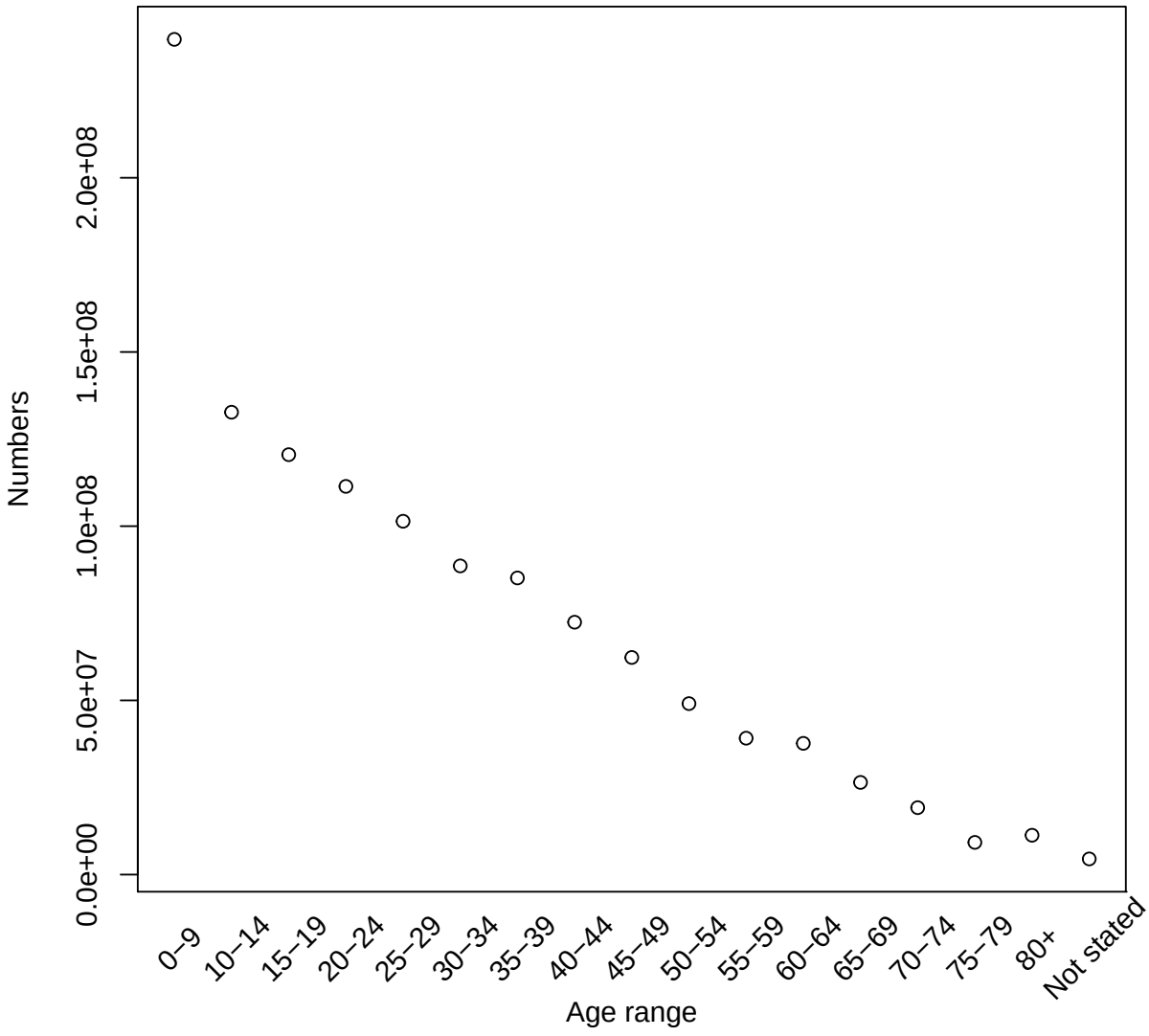
1. Just use `read.csv` without the skip options and see what happens.
2. There are some other methods suggested online for skipping the sixth line – which look less clunky – but did not work for me! Try some of these and let us know if any other trick works!

Now that we have the data, let us start the descriptive statistics!

1 The graphical measures!

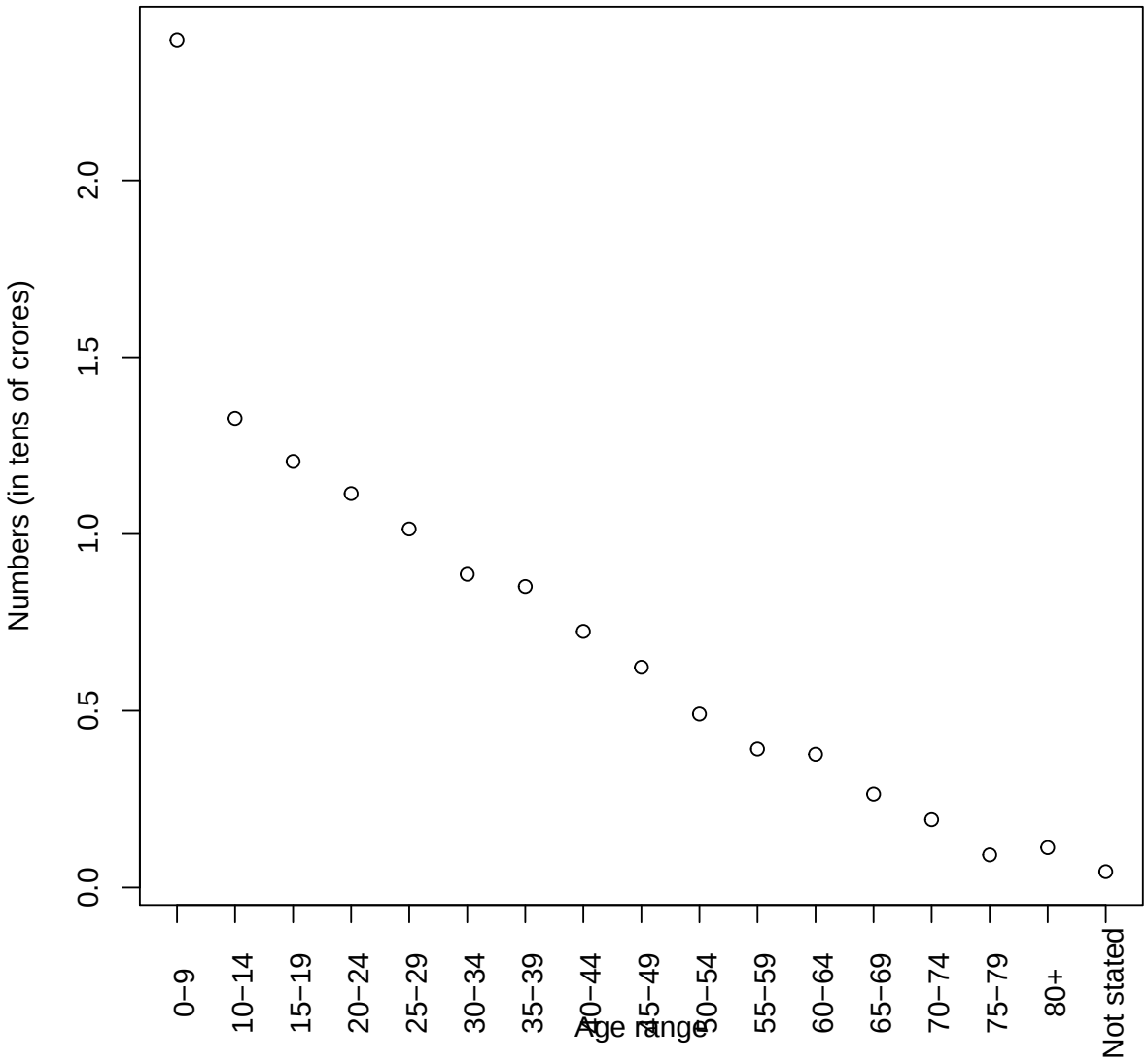
The first and easiest thing to do is to plot the data. For example, let us plot the age versus the number of persons – for all of India. So, basically, we want to plot columns X1 and X2 and the rows 2 to 18. But, X1 is not numbers – it is a mix of range of numbers, and characters. So, we are going to prepare a list of labels, plot X2 without labeling x axis, and then identify the positions for the x-label, and at the x-label positions, pick the name from the label list.

```
label_list <- c("0-9","10-14","15-19","20-24","25-29","30-34",
                "35-39","40-44","45-49","50-54","55-59","60-64",
                "65-69","70-74","75-79","80+","Not stated")
plot(X$X2[2:18],xaxt="n",xlab="Age range",ylab="Numbers")
axis(1,at=X$X2[2:18],labels=FALSE)
text(seq(1, 17, by=1), par("usr")[3] - 0.2, labels = label_list,
     srt = 45, offset = 1.5, pos = 1, xpd = TRUE)
```



We are still not happy because the numbers are very large and are not clearly seen. So, let us report the numbers in tens of crores.

```
label_list <- c("0-9","10-14","15-19","20-24","25-29","30-34",
                "35-39","40-44","45-49","50-54","55-59","60-64",
                "65-69","70-74","75-79","80+","Not stated")
y <- X$X2[2:18]*10^{-8}
plot(y,xaxt="n",xlab="Age range",
     ylab="Numbers (in tens of crores)")
axis(1,at=seq(1,17, by= 1),labels=FALSE)
text(seq(1,17, by= 1), par("usr")[3]-0.2,
     labels = label_list, srt = 90, offset = 0.5,
     pos = 1,xpd=TRUE)
```



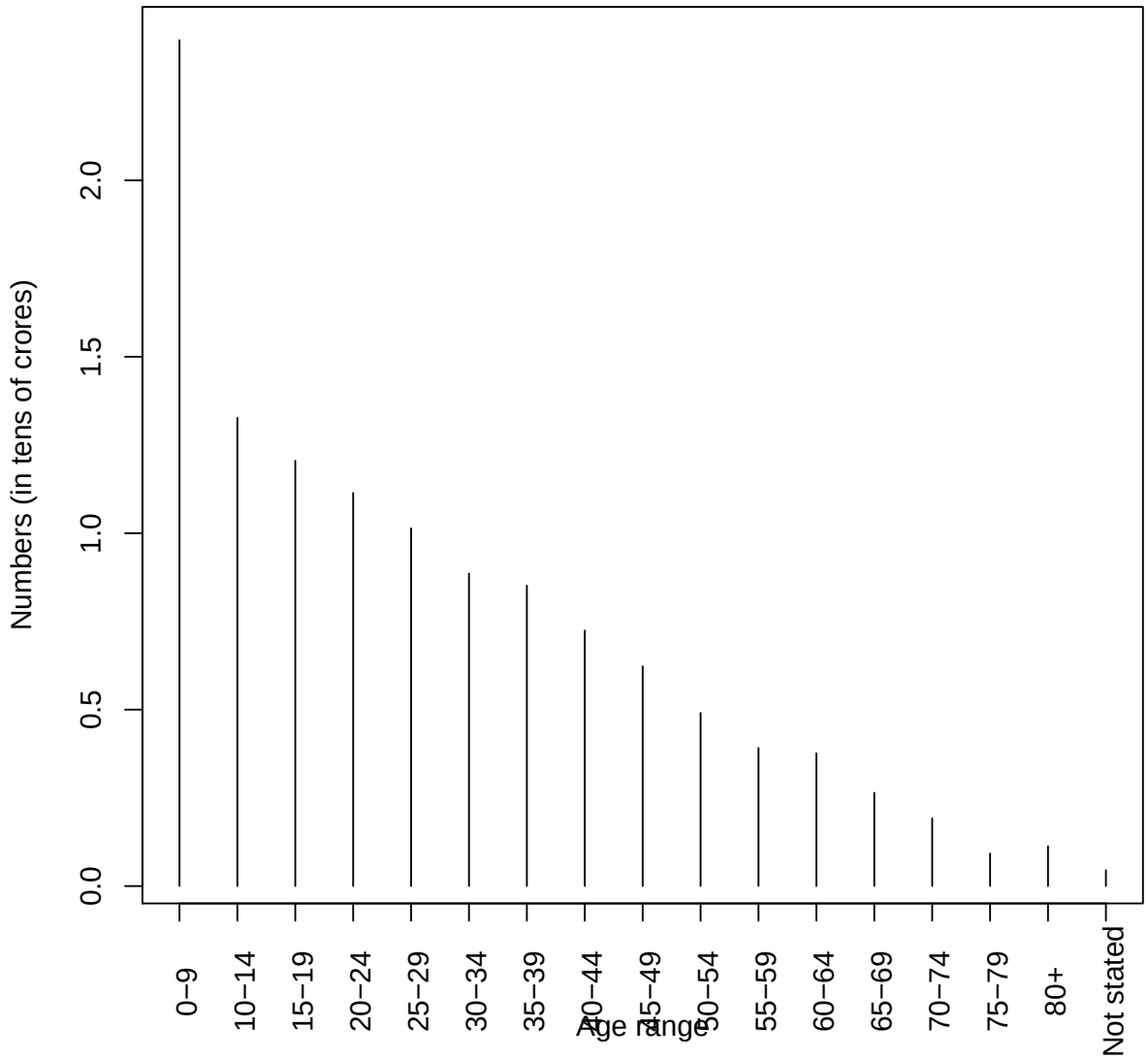
We have used two new commands in addition to plot –`axis` and `text`. Please peruse the help files to understand the parameters we have set as well as the variables. Another way to understand these parameters is to run the script by changing the parameter value or by removing the parameter and seeing the effect of the same on the plot; if this does not result in an error message, there can be lots of learning in this trial-and-error process!!

Can you guess the mean age of the Indian population from this plot?

Let us try one more type of plotting – this time around with bars!

```
label_list <- c("0-9","10-14","15-19","20-24","25-29","30-34",
                "35-39","40-44","45-49","50-54","55-59","60-64",
                "65-69","70-74","75-79","80+","Not stated")
y <- X$X2[2:18]*10^{-8}
plot(y,type="h",xaxt="n",xlab="Age range",
```

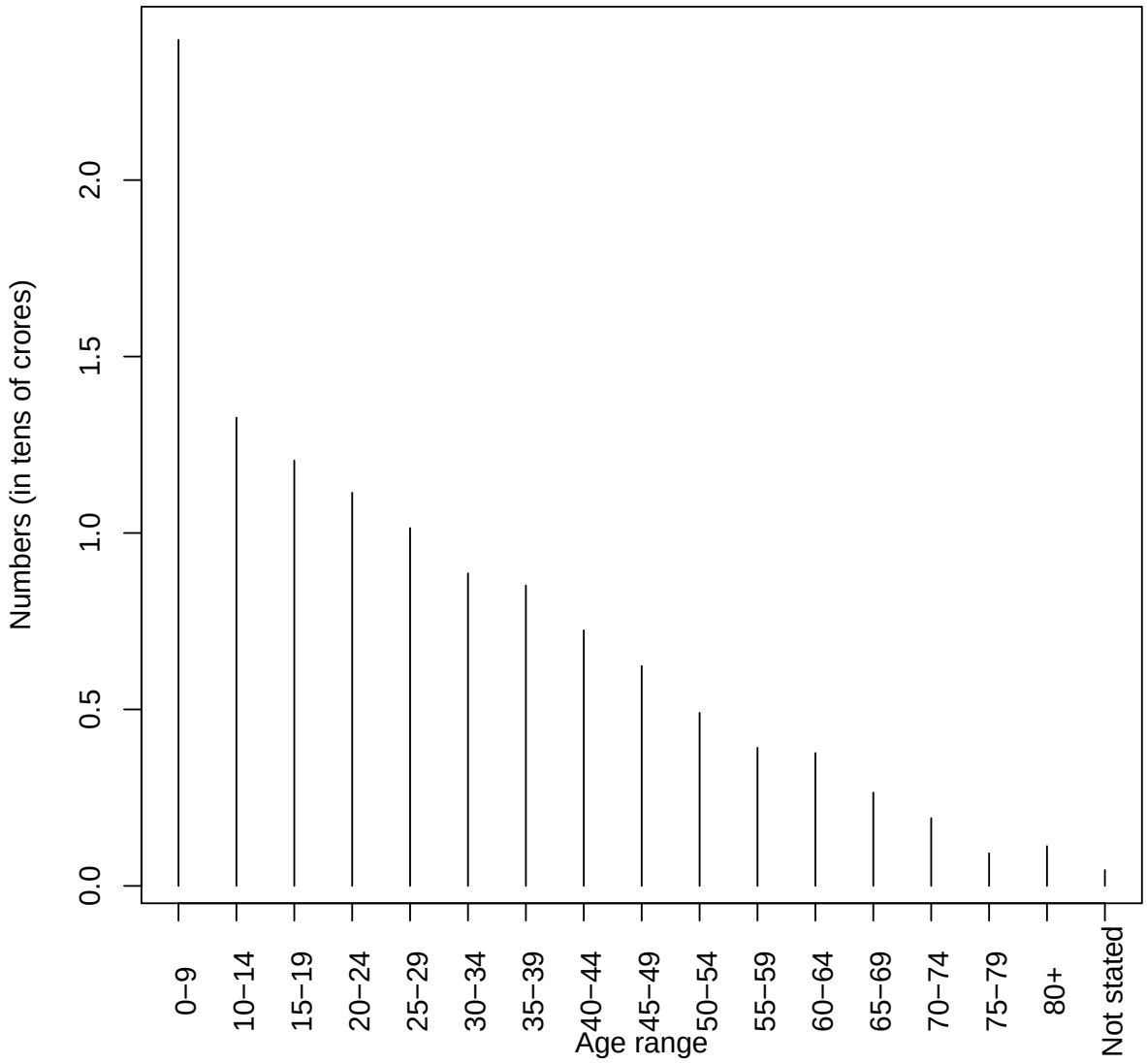
```
     ylab="Numbers (in tens of crores)")
axis(1,at=seq(1,17, by= 1),labels=FALSE)
text(seq(1,17, by= 1), par("usr")[3]-0.2, labels = label_list,
     srt = 90, offset = 0.5, pos = 1,xpd=TRUE)
```



In this plot, the x label is getting overwritten by the x-tick marks; let us fix that.

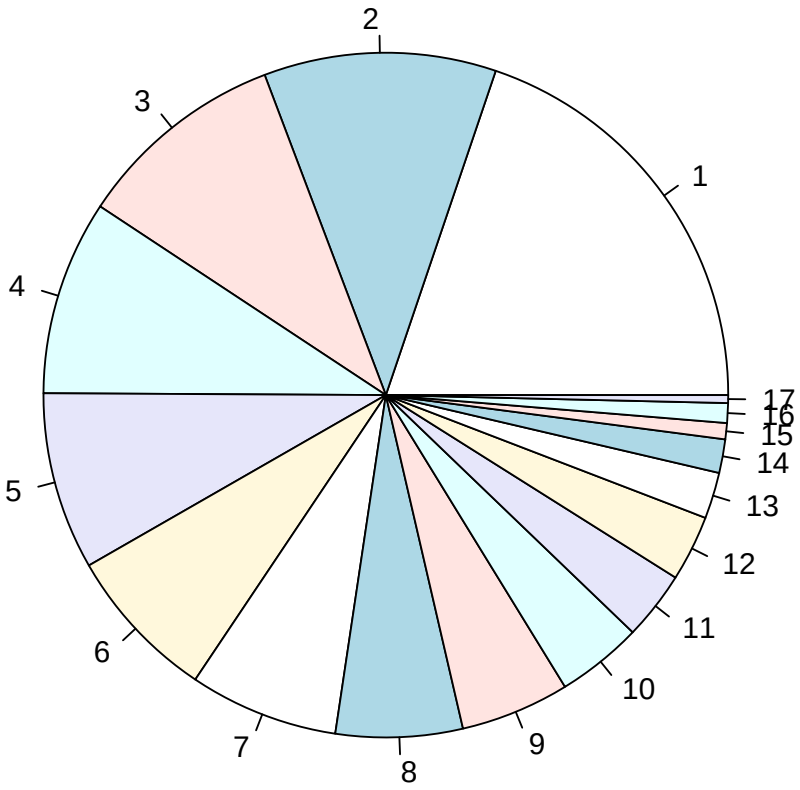
```
label_list <- c("0-9","10-14","15-19","20-24","25-29","30-34",
                "35-39","40-44","45-49","50-54","55-59","60-64",
                "65-69","70-74","75-79","80+","Not stated")
y <- X$X2[2:18]*10^{-8}
plot(y,type="h",xaxt="n",xlab="",
     ylab="Numbers (in tens of crores)")
axis(1,at=seq(1,17, by= 1),labels=FALSE)
```

```
text(seq(1,17, by= 1), par("usr")[3]-0.2, labels = label_list, srt = 90, offset = 0.5, pos = 1,xpd=TRUE)
title(xlab="Age range",line=3.5)
```



Let us make a pie chart of the data:

```
pie(y)
```



Home work

Can you make the labels of the pie-chart to read the age range instead of numbering them from 1 to 17?

Let us make a stem-and-leaf plot of the data

```
stem(y,scale=1)
```

```
##
##   The decimal point is at the |
##
##   0 | 0112344
##   0 | 56799
##   1 | 0123
##   1 |
```

```
##   2 | 4
```

Home work

Compare the plot with the data itself. What do you see? How about stem plot with the option scale = 2? What happens to stem plot of the Grandmother's tale data with scales 1 and 2? Compare and understand.

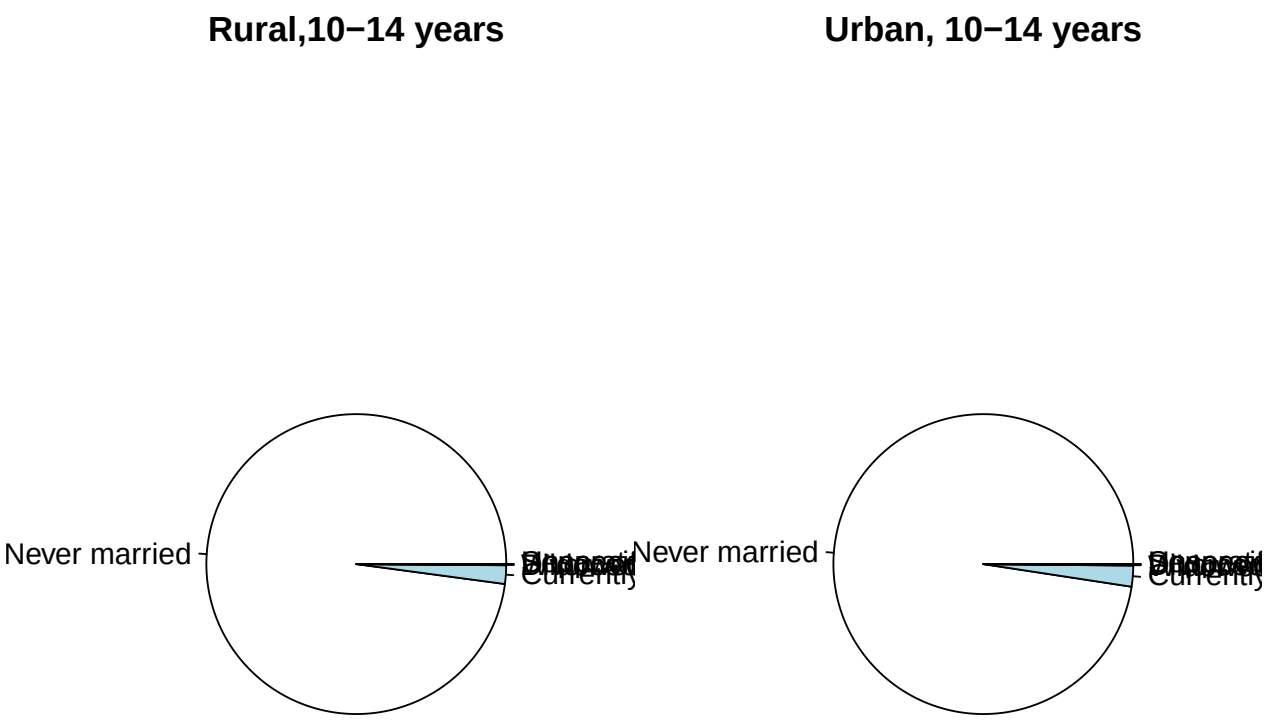
Of course you can also plot the dot chart, histogram and box plot. Try them and see what they mean!

1.1 Comparisons

Note that in this data, what is interesting is the break-up of each age group in terms of their marriage status. We have not yet plotted any of it! Let us make some plots of that type!

To begin with, let us compare the marital status, for persons in the age group 10 to 14, in rural and urban areas. That is, we are interested in plotting information from two different rows and comparing them! How do we do this?

```
R <- subset(X,X1=="10-14" & X.4=="Rural" & X.3=="INDIA")
U <- subset(X,X1=="10-14" & X.4=="Urban" & X.3=="INDIA")
r <- c("Never married"=R$X5,"Currently married"=R$X8,
       "Widowed"=R$X11,"Separated"=R$X14,
       "Divorced"=R$X17,"Unspecified"=R$X20)
u <- c("Never married"=U$X5,"Currently married"=U$X8,
       "Widowed"=U$X11,"Separated"=U$X14,
      "Divorced"=U$X17,"Unspecified"=U$X20)
par(mfrow=c(1,2))
pie(r,main="Rural,10-14 years")
pie(u,main="Urban, 10-14 years")
```



The first two lines pull out the relevant data for rural and urban regions. The next two lines pull out the data that we want to plot. The last three lines are the pie charts themselves.

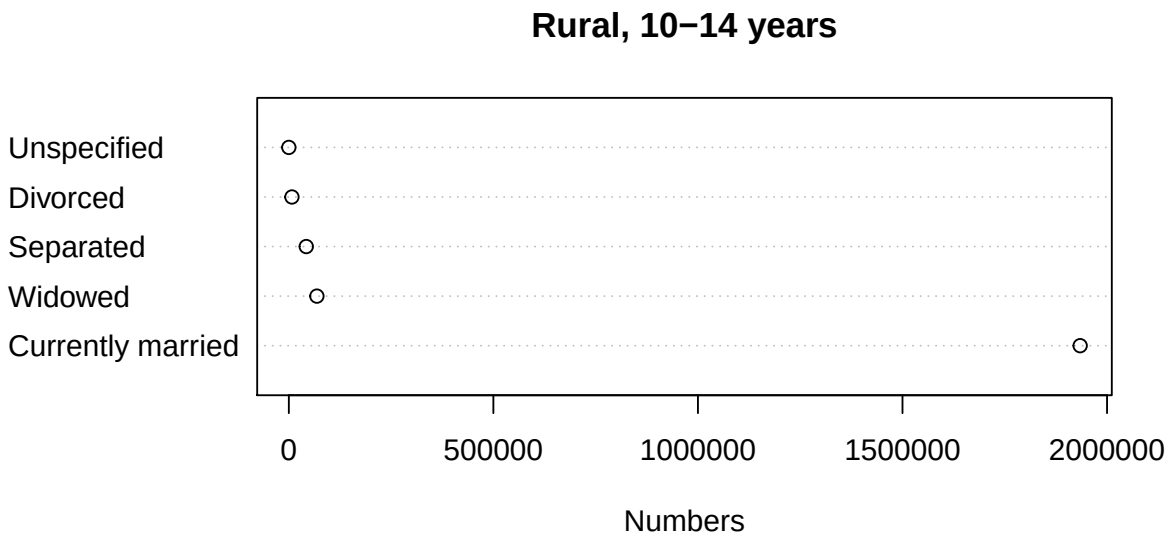
It is clear that the pie chart is not very useful in this case – except to indicate that child marriage seems to be prevalent. It looks like the prevalence is the same in both urban and rural areas – as can be seen from the following calculation, even though in terms of absolute numbers, the rural numbers are much higher, as a fraction of the population in the age group, if anything, the urban prevalence is a bit high.

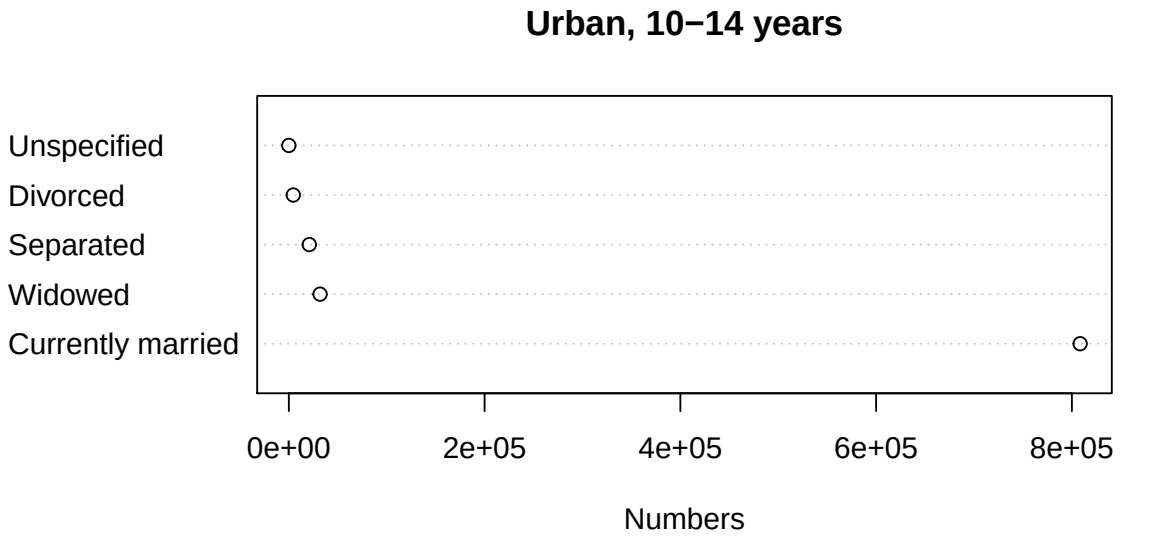
```
sum(r[2:5])

## [1] 2052981

sum(u[2:5])

## [1] 865793
```

```r
sum(r[2:5])/R$X2

## [1] 0.0212075

sum(u[2:5])/U$X2

## [1] 0.02411363
```

A much better comparison can be achieved using a dot chart as follows:

```r
par(mfrow=c(2,1))
dotchart(r[2:6],main="Rural, 10-14 years",xlab="Numbers")
dotchart(u[2:6],main="Urban, 10-14 years",xlab="Numbers")
```





You can understand this data much better, especially, if you pay attention to the numbers on the x-axis.

Home work

- Do a comparison between rural and urban population in the age bracket 20-24 for all of India – using both pie charts and strip charts.
- Do a comparison between rural Kerala and urban Tamilnadu population in the age bracket 30-34 – using both pie charts and strip charts.
- Do a comparison between rural and urban population in the age bracket 40-44 – using both pie charts and strip charts – for a state of your choice.
- Do a comparison between population in the age brackets 20-24, 30-24 and 40-44 of Maharashtra – using both pie charts and strip charts. Note that the number of plots now are three instead of 3.
- Do a comparison between male and female population in the age brackets 25-29 of Haryana – using both pie charts and strip charts. How does this data compare with the male and female population in the same age bracket from Manipur?
- Can you identify the state in which the child marriage is the most prevalent? Note that in this and some of the other questions, it is important to normalise by the total population – absolute numbers by themselves do not mean much; only the percentage of the population makes sense for comparisons.
- Can you identify the state in which the child marriage of the girl child is the most prevalent? Is there any state in which child marriage is more prevalent among boys than among girls?
- Of course, the possibilities are endless! So, keep playing with the data!

Let us proceed to the calculation of some numerical descriptors of the data.

2 Numerical description

One can proceed to the calculation of some numerical descriptors of the data. Of course, if you feed the data to the commands that we learnt, such as `mean, median, sd, var` and so on, you will get the numbers. However, in this case, these numbers do not make much sense. (Why?)

One can calculate the mean age from the data; but, as you can see, the data is already binned in terms of age. This is not uncommon. Many a times we get binned data and we have to have the means of getting some of these numerical descriptors from them. One way to calculate the mean is to take a weighted average – for example, taking all the persons in the age range 0-9, and multiply that with 4.5 (which is the mean age of the range) and so on, add all of them and divide by the total number. Even then, there is an issue. How do we account for the data binned as 80+? Can we assume it is between 80 and 100 and take 90 as the mean age? How much does your answer change if you keep it at 100 (assuming 120 as the absolute upper limit)?

So, I will stop this tutorial at this point, and will return with the next data set, namely, rainfall data in our next session!