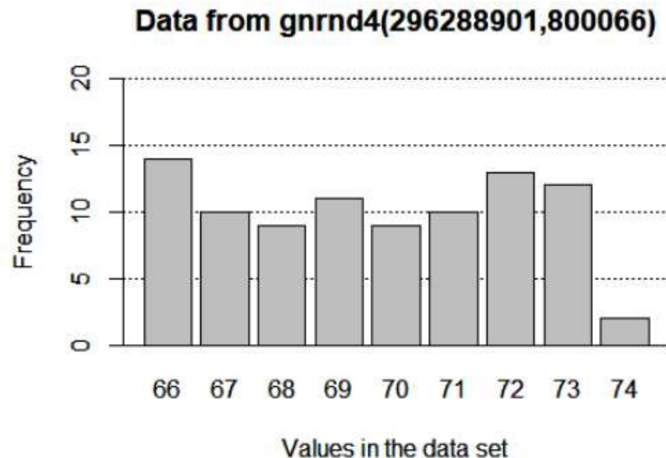
Topic 7k: Frequency Tables in R

Suppose we have the following data. We want to generate a frequency table for this.

gnrnd4(296288901, 800066)

70	72	68	67	69	72	71	69	73	71	66	68	71	66	73	73	70	68	72	68	71	70	66	73
70	73	71	70	66	67	66	68	69	72	73	69	73	69	68	66	66	69	71	69	67	66	67	66
73	66	72	67	66	72	74	67	68	67	71	72	66	67	67	67	68	72	71	66	69	70	72	66
73	71	70	70	72	70	69	73	69	73	72	74	69	73	72	72	68	71						

Assuming that we generate the data using gnrnd4(296288901, 800066) to get those values into L1, then we know how to get a picture of the frequency of each value in the data by using the barplot(table(L1)) function. Here is an example of such a picture, with a few enhancements.



Remember that we used table(L1) to get the frequency of each distinct value in the data. In fact, if we just use the command table(L1) we get the console output shown at the right.

```
> table(L1)
L1
66 67 68 69 70 71 72 73 74
14 10  9 11  9 10 13 12  2
```

```
26       # now, rather than re-enter the frequencies we
27       # will have R compute them again and store them
28       # in a variable we will call   freqs
29  freqs <- table(L1)
```

Then, we can compute the relative frequency for each value by taking the frequency and dividing it by the total number of data points, in this case 90. That gives us a new table.

```
30       # to compute the relative frequency we divide
31       #  the frequencies by the total number of items
32  total <- length(L1)
33  rel_freq <- freqs/total
34  rel_freq
```

```
> rel_freq
L1
        66         67         68         69         70
0.15555556 0.11111111 0.10000000 0.12222222 0.10000000
        71         72         73         74
0.11111111 0.14444444 0.13333333 0.02222222
```

Another way to organize our view of the frequencies is to compute the cumulative frequency. That means that we find the "running" total of the frequencies as we read across the table.

```
35       # to compute the cumulative frequencies we
36       # use the cumsum() function
37  cum_count <- cumsum( freqs)
38  cum_count
```

```
> # to compute the cumulative frequencies we
>      # use the cumsum() function
> cum_count <- cumsum( freqs)
> cum_count
66 67 68 69 70 71 72 73 74
14 24 33 44 53 63 76 88 90
```

We expand the table again by adding a row for the cumulative relative frequency which is the same as the relative cumulative frequency. That is, the new row can be computed by getting the "running" total of the relative frequencies, or by dividing the cumulative frequencies by 80. We get the same value either way.

```
39      # to compute the cumulative relative
40      # frequencies we just divide the cumulative
41      # frequencies by the total number of items
42  cum_rel_freq <- cum_count/total
43  cum_rel_freq
```

```
> # to compute the cumulative relative
>      # frequencies we just divide the cumulative
>      # frequencies by the total number of items
> cum_rel_freq <- cum_count/total
> cum_rel_freq
       66        67        68        69        70 
0.1555556 0.2666667 0.3666667 0.4888889 0.5888889 
       71        72        73        74 
0.7000000 0.8444444 0.9777778 1.0000000 
```

As long as we are doing these computations we might as well add another easy one, namely, we can compute the number of degrees to allocate in a pie chart to each of the different data values. To do this we merely multiply the relative frequency by 360.

```
44      # to compute the degrees to allocate in a pie
45      # chart we just multiply the relative frequency
46      # times 360
47  deg_pie <- 360*rel_freq
48  deg_pie
```

```
> # to compute the degrees to allocate in a pie
>      # chart we just multiply the relative frequency
>      # times 360
> deg_pie <- 360*rel_freq
> deg_pie
L1
66 67 68 69 70 71 72 73 74 
56 40 36 44 36 40 52 48  8 
```

But, rather than doing all of those steps each time we come across this kind of task we can place those steps into a function that we just need to load into our environment and then run. make_freq_table() is such a function.

```
58  source("../make_freq_table.R")
59  make_freq_table( L1 )
```

```
> source("../make_freq_table.R")
> make_freq_table( L1 )
  Items Freq   rel_freq cumul_freq rel_cumul_freq pie
1    66   14 0.15555556         14      0.1555556  56
2    67   10 0.11111111         24      0.2666667  40
3    68    9 0.10000000         33      0.3666667  36
4    69   11 0.12222222         44      0.4888889  44
5    70    9 0.10000000         53      0.5888889  36
6    71   10 0.11111111         63      0.7000000  40
7    72   13 0.14444444         76      0.8444444  52
8    73   12 0.13333333         88      0.9777778  48
9    74    2 0.02222222         90      1.0000000   8
```