Recall

Recall week1

R- c function

Suppose we wish to enter attendance on 10 days in W.O.M. 2018 class.

40, 39, 15, 6, 18, 22, 30, 21, 15, 23

```
> attendance = c(40, 39, 15, 6, 18, 22, 30, 21, 15, 23)
```

R- c function

The output should be

```
> attendance = c(40, 39, 15, 6, 18, 22, 30, 21, 15, 23)
```

In the above, we have assigned values to a variable called `attendance`.

The assignment operator is `=`.

The values do not get displayed automatically unless we call it with `attendance` as below.

```
> attendance = c(40, 39, 15, 6, 18, 22, 30, 21, 15, 23)
> attendance
 [1] 40 39 15  6 18 22 30 21 15 23
```

R-inbuilt functions

We also discussed that there were in-built functions R.

```
> meandirectly = (40+ 39+ 15+ 6+ 18+ 22+ 30+ 21+ 15+ 23)/10
> meandirectly

[1] 22.9

> meaninbuilt = mean(attendance)
> meaninbuilt

[1] 22.9
```

R-slicing

Changing one element of attendance: Suppose we want to change the entry on day 4 from 6 to 16.

```
> attendance

 [1] 40 39 15  6 18 22 30 21 15 23

> attendance2 = attendance # create a copy of attendance
> attendance2[4] = 16
> attendance2

 [1] 40 39 15 16 18 22 30 21 15 23
```

R-slicing

Selecting few elements of attendance: Suppose we want to see attendance on day 1, 3, 5. We can do this using the c function

```
> attendance

 [1] 40 39 15  6 18 22 30 21 15 23

> attendance[3] # gives only 3rd day

[1] 15

> attendance[c(1,3,5)]

[1] 40 15 18
```

command in R tells R to ignore the rest of the line. We can use this to have comments on commands for our future reference.

R- Logical Operators ==, <=, >=, <, >

Selecting few elements of attendance: Suppose we want to see on what days was attendance equal to 30 and days lesser than or equal to 20.

```
> attendance

 [1] 40 39 15  6 18 22 30 21 15 23

> y = which(attendance == 30)
> y

[1] 7

> z = which(attendance <= 20)
> z

[1] 3 4 5 9
```

R- Creating sequence of vectors

```
> x = 1:100
> x

  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 [91]  91  92  93  94  95  96  97  98  99 100

> x[x < 10 | x > 90]

 [1]   1   2   3   4   5   6   7   8   9  91  92  93  94  95  96  97  98  99 100
```

Data

- Data and its analysis has a rich and wide literature.
- In this course we will try understand Data using Statistical Inference.
- Three kinds of Data:
  - Categorical Data
  - Discrete Numeric Data
  - Continuous Numeric Data
- *On the Theory of Scales of Measurement* By S. S. Stevens Science 07 Jun 1946: Vol. 103, Issue 2684, pp. 677-680, gave a broad classification of data from measurements into 9 categories.

Categorical Data

- Data that Records categories (either numeric or character(s)).
- It is used to classify data.
- Bar Charts are used to visualise such data

Bar Charts for Categorical Data

- A bar chart is a graph where for each category a bar with a height proportional to the count in the table is drawn.

- Along x-axis the categories (or levels) are displayed.

Categorical Data

A survey is conducted in B.Math (hons.) IInd year class to check on their thirst quenching fresh fruit juice preference:

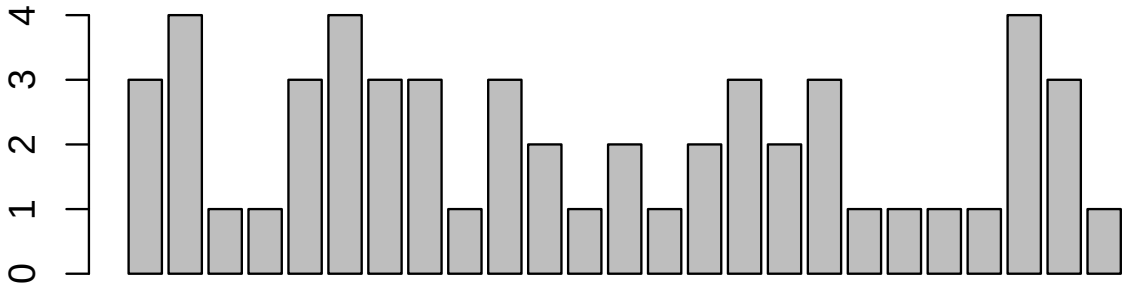
1: Orange, 2: Water Melon, 3: Pineapple, 4: Apple.

The results are recorded below.

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)
> x
 [1] 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
```

Bar Charts -Raw Data- Categorical Data

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)

>  barplot(x)
```

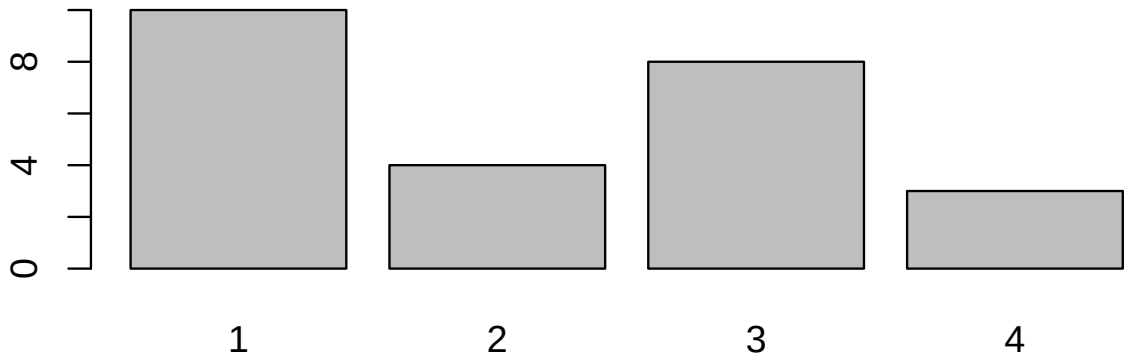


- It is not particularly discerning.
- Treating everyone as separate catgeory is resulting in too many categories and clearly non-informative.

Bar Charts - Frequencies - Categorical Data

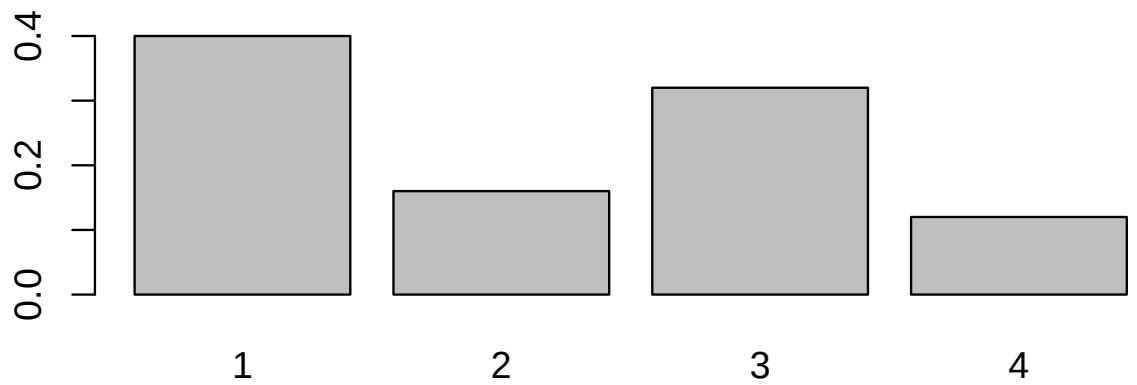
- We can first summarize the data.
- `table` command provides the frequency of each unique value of the data.

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)
> table(x)

x
 1  2  3  4
10  4  8  3

> barplot(table(x))
```



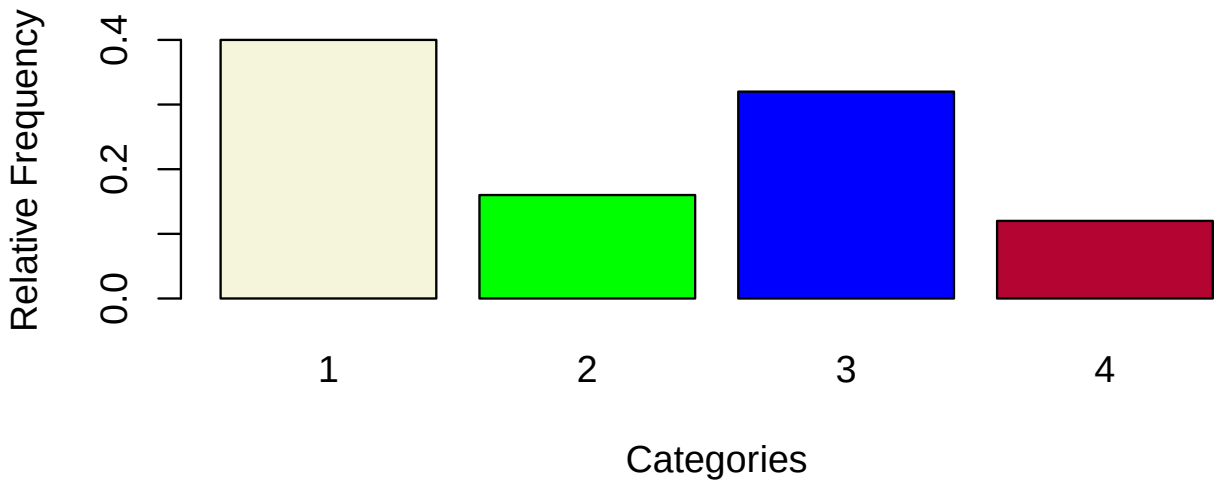
Bar Charts - Proportions - Categorical Data

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)
> x
 [1] 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
> length(x)
[1] 25
> table(x)
x
 1  2  3  4
10  4  8  3
> z = table(x)/length(x)
> barplot(z)
```



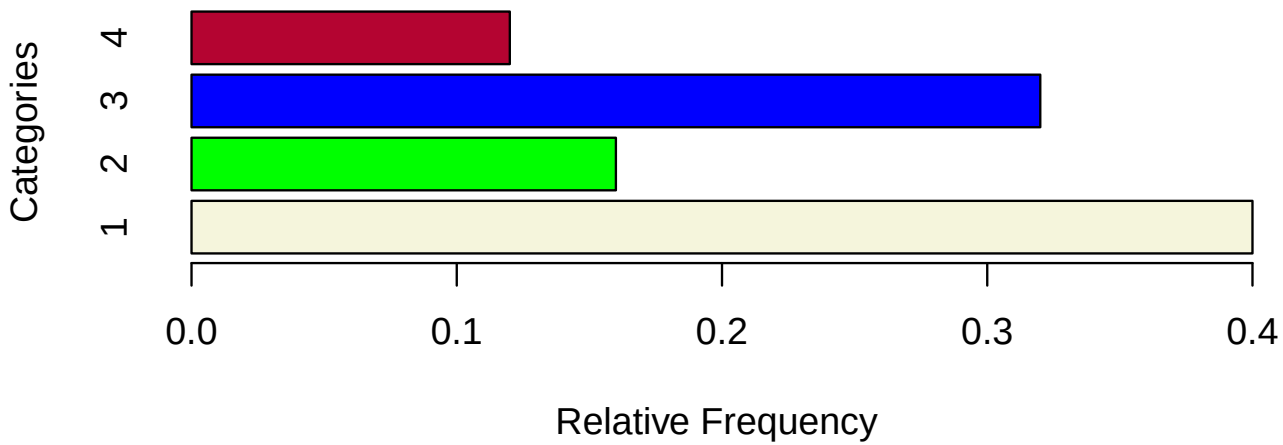
Bar Charts - Proportions - Categorical Data - Color

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)

> z = table(x)/length(x)
> barplot(z, xlab = "Categories",  ylab = "Relative Frequency",col = c("beige","Green","Blue","#B40431"))
```



Bar Charts –Horizontal

```
> x = c(3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1)

> z = table(x)/length(x)
> barplot(z,horiz=TRUE,ylab="Categories",xlab="Relative Frequency",col=c("beige","Green","Blue","#B40431")
```



Categorical Data

```
> x = c("TRUE", "FALSE","FALSE", "TRUE","TRUE","FALSE","FALSE","FALSE")
> x

[1] "TRUE"  "FALSE" "FALSE" "TRUE"  "TRUE"  "FALSE" "FALSE" "FALSE"

> factor(x)

[1] TRUE  FALSE FALSE TRUE  TRUE  FALSE FALSE FALSE
Levels: FALSE TRUE
```

- Character data is created by just giving values within quotes.
- For modeling functions to work with categorical data correctly the data is usually stored as factors.
- `factor(x)` is made from character data provide by `x` and is stored as a vector of integer values with a corresponding set of character values to use when the factor is displayed.

Read more above Factors at
https://www.stat.berkeley.edu/ s133/factors.html

Categorical Data

```
> x = c("TRUE", "FALSE","FALSE", "TRUE","TRUE","FALSE","FALSE","FALSE")
> x

[1] "TRUE"  "FALSE" "FALSE" "TRUE"  "TRUE"  "FALSE" "FALSE" "FALSE"

> factor(x)

[1] TRUE  FALSE FALSE TRUE  TRUE  FALSE FALSE FALSE
Levels: FALSE TRUE
```

- R has two classes for categorical data :*Factor* and *Character*
- Both numeric and character variables can be made into factors, but a factor's levels will always be character values.
- The `Levels` of a factor are a list of all possible categories for the data in factor.
- One can use the levels in R to order the factors.

Categorical Data

```
> x = c("TRUE", "FALSE","FALSE", "TRUE","TRUE","FALSE","FALSE","FALSE")
> x

[1] "TRUE"  "FALSE" "FALSE" "TRUE"  "TRUE"  "FALSE" "FALSE" "FALSE"

> table(x)

x
FALSE  TRUE
    5     3
```