

## (§1.2) Descriptive statistics; (§1.3) Measures of location

Common naming conventions:

- Population size:  $N$
- Sample size:  $n$
- Sample from two different populations:  $n, m$ , or  $n_1, n_2$
- Data:  $x_1, x_2, x_3, \dots, x_n$

### Stem-and-leaf displays

```
> x = sample(1:50, size=20, replace=TRUE)
> sort(x)
[1] 2 2 2 3 9 14 18 19 20 21 21 22 22 29 30 32 32
[18] 33 44 47
> stem(x)
```

The sample function generates numbers in the range provided as the first argument, with a size equal to the second argument. sort(x) sorts the values stored in x, and stem(x) does the following:

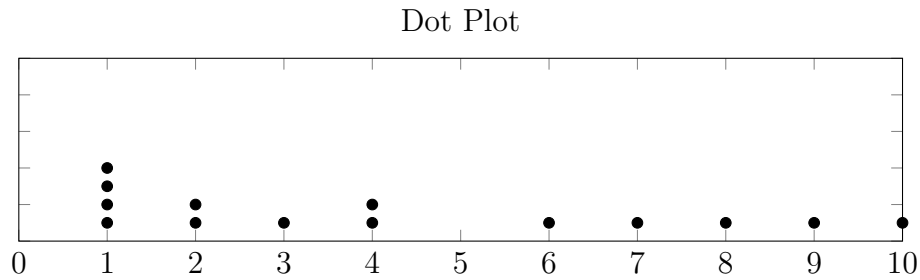
Each “stem” refers to the highest digits and each “leaf” is the latter digits. This is the stem-and-leaf display for the dataset stored in x:

Stem	Leaves
0	2 2 2 3 9
1	4 8 9
2	0 1 1 2 2 9
3	0 2 2 3
4	4 7

### Dot plots

```
> x <- sample(1:10, size=15, replace=TRUE)
> x
[1] 7 8 1 3 4 10 1 2 2 1 1 4 4 9 6
> stripchart(x, method="stack", offset=0.5, at=0.15, pch=20)
```

`stripchart()` constructs the dot plot. The `offset` and `at` values affect the appearance. `pch` means `p`rinting `ch`aracter. Having it set to 20 makes the dot solid.



## Histograms

- For quantitative data.
- May differ depending on whether a variable is discrete or continuous.

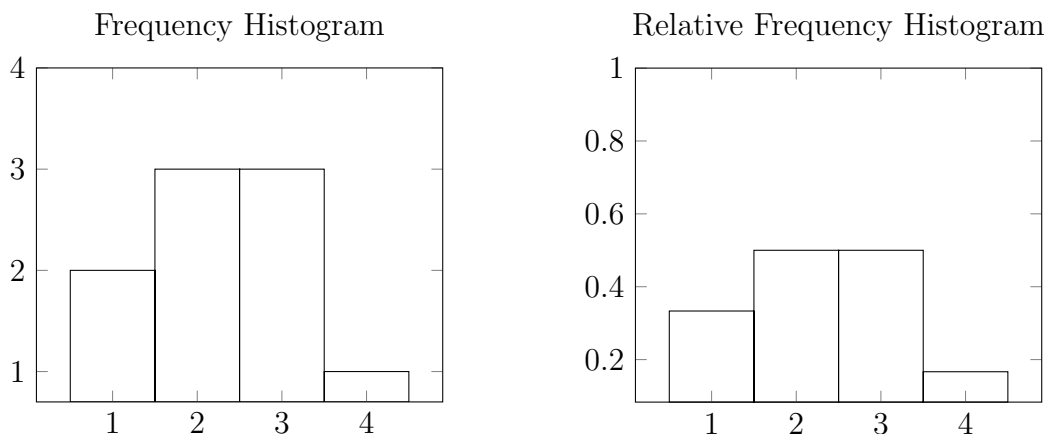
**Discrete** data is either infinite or countably infinite: orderable, typically integers. Includes ration numbers.

- $x_1 = \#$  of tuberculosis cases in the 50 largest U.S. cities in 2013.  $\leftarrow$  Might be 0, 1, 2, ... but  $\neq \infty$ .
- $x_i$  # of “successes” when 100 patients treated.  $\leftarrow$  Count data in which we know a hard upper limit (in this case 100).
- $x_i$  # of rooms in a rental unit.  $\leftarrow$  1, 2, ..., 100. At least 1 guaranteed as a hard minimum limit.

Histograms, when data is discrete, use vertical bars with height proportional to the number of times each value occurs.

### Frequency Histogram vs Relative Frequency Histogram

For the data:  $\{1, 1, 2, 2, 2, 3\}$



Relative Frequency Histograms have a total height of 1.

## R Code for Histograms

```
> # Use R to simulate 100 values from {1, 2, 3, ..., 10}
> my_data <- sample(1:10, size=100, replace=TRUE)
> # Here are the first 20 values:
> my_data[1:20]
[1] 4 3 5 6 5 4 10 2 6 2 5 8 7 10 4 10 7 6
[19] 5 7
> # Create a frequency histogram:
> hist(my_data, main="Plot_1_-_discrete_data")
> # Create a relative frequency histogram:
> hist(my_data, probability=TRUE, main="Plot_2")
```

*Note: R uses # for comments. You're a CS major, you know what these functions and their arguments do.*

## R Code of a Histogram Using Continuous Data

```
> # Generate 100 rational values from 1:10.
> my_data <- runif(100, min=1, max=10)
> # Create a histogram using the same code for discrete data:
> hist(my_data, main="Plot_3_-_Continuous_Data")
> # And a relative frequency histogram:
> hist(my_data, probability=TRUE, main="Plot_4")
```

Here, `r` refers to random and `unif` refers to uniform in `runif`, it does not refer to `runif`.

Endpoint: 41:32.