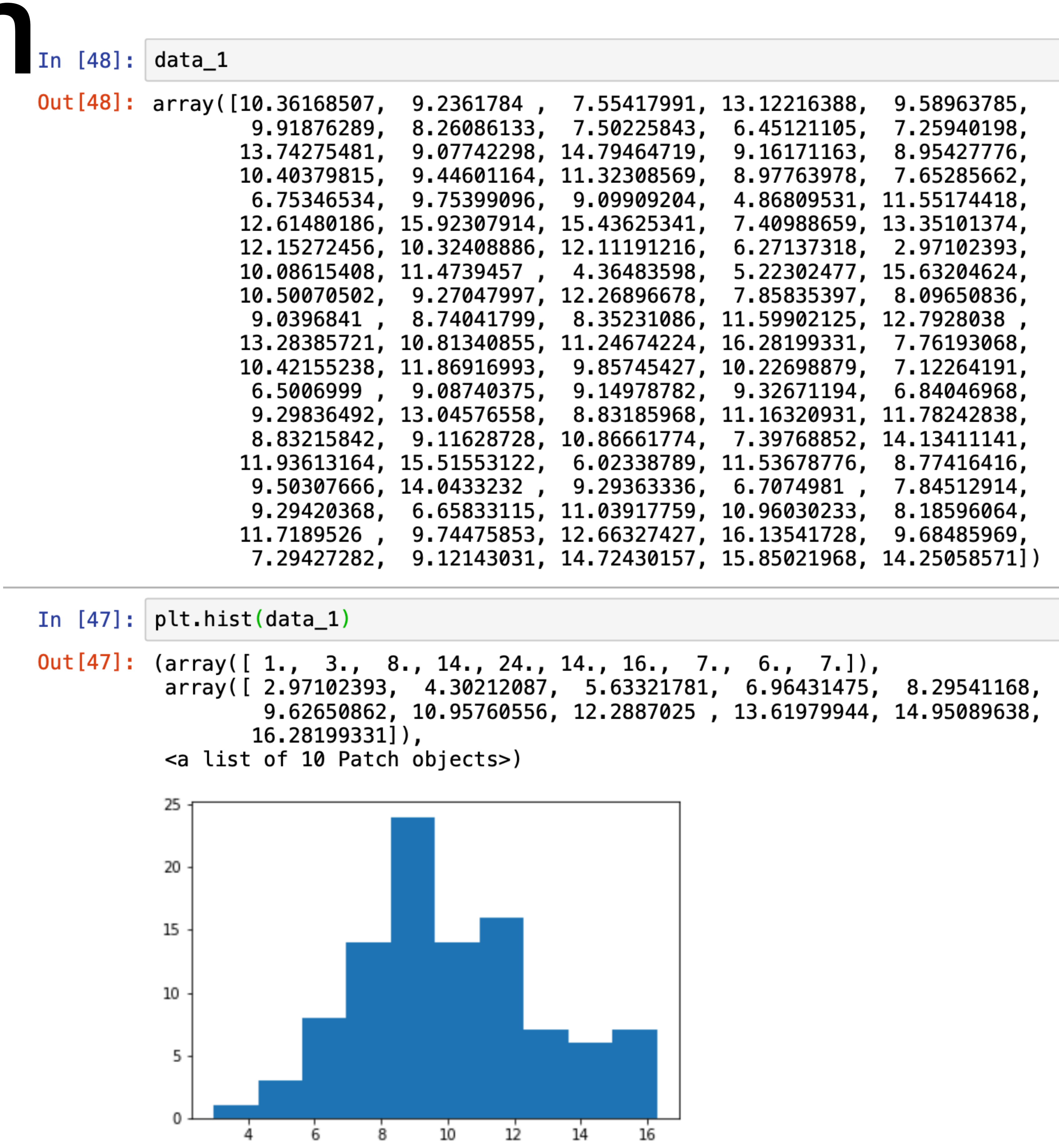
Introduction to Scientific Computing I

Lecture 15

Amir Farbin

Histogram



```
In [48]: data_1

Out[48]: array([10.36168507,  9.2361784 ,  7.55417991, 13.12216388,  9.58963785,
                 9.91876289,  8.26086133,  7.50225843,  6.45121105,  7.25940198,
                13.74275481,  9.07742298, 14.79464719,  9.16171163,  8.95427776,
                10.40379815,  9.44601164, 11.32308569,  8.97763978,  7.65285662,
                 6.75346534,  9.75399096,  9.09909204,  4.86809531, 11.55174418,
                12.61480186, 15.92307914, 15.43625341,  7.40988659, 13.35101374,
                12.15272456, 10.32408886, 12.11191216,  6.27137318,  2.97102393,
                10.08615408, 11.4739457 ,  4.36483598,  5.22302477, 15.63204624,
                10.50070502,  9.27047997, 12.26896678,  7.85835397,  8.09650836,
                 9.0396841 ,  8.74041799,  8.35231086, 11.59902125, 12.7928038 ,
                13.28385721, 10.81340855, 11.24674224, 16.28199331,  7.76193068,
                10.42155238, 11.86916993,  9.85745427, 10.22698879,  7.12264191,
                 6.5006999 ,  9.08740375,  9.14978782,  9.32671194,  6.84046968,
                 9.29836492, 13.04576558,  8.83185968, 11.16320931, 11.78242838,
                 8.83215842,  9.11628728, 10.86661774,  7.39768852, 14.13411141,
                11.93613164, 15.51553122,  6.02338789, 11.53678776,  8.77416416,
                 9.50307666, 14.0433232 ,  9.29363336,  6.7074981 ,  7.84512914,
                 9.29420368,  6.65833115, 11.03917759, 10.96030233,  8.18596064,
                11.7189526 ,  9.74475853, 12.66327427, 16.13541728,  9.68485969,
                 7.29427282,  9.12143031, 14.72430157, 15.85021968, 14.25058571])
```

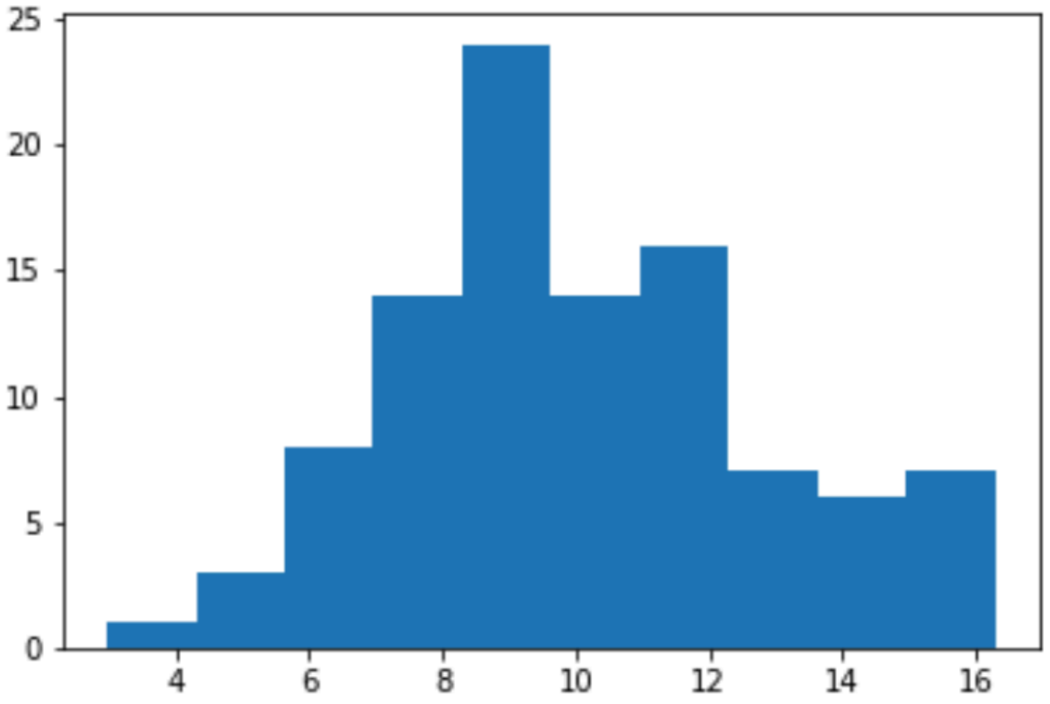
```
In [47]: plt.hist(data_1)

Out[47]: (array([ 1.,  3.,  8., 14., 24., 14., 16.,  7.,  6.,  7.]),
          array([ 2.97102393,  4.30212087,  5.63321781,  6.96431475,  8.29541168,
                  9.62650862, 10.95760556, 12.2887025 , 13.61979944, 14.95089638,
                 16.28199331]),
          <a list of 10 Patch objects>)
```

```
        7.29427282,  9.12143031, 14.72430157, 15.85021968, 14.25058571])
```

```
In [47]: plt.hist(data_1)
```

```
Out[47]: (array([ 1.,  3.,  8., 14., 24., 14., 16.,  7.,  6.,  7.]),
          array([ 2.97102393,  4.30212087,  5.63321781,  6.96431475,  8.29541168,
                  9.62650862, 10.95760556, 12.2887025 , 13.61979944, 14.95089638,
                 16.28199331]),
          <a list of 10 Patch objects>)
```



Histogram

In Lab 4 you are asked to write a histogram function:

- User inputs a list of values `x` and optionally `n_bins` which defaults to 10.
- If not supplied, find the minimum and maximum (`x_min` , `x_max`) of the values in x.
- Determine the bin size (`bin_size`) by dividing the range of the function by the number of bins.
- Create an empty list of zeros of size `n_bins` , call it `hist` .
- Loop over the values in `x`
  - Loop over the values in `hist` with index `i` :
    - If x is between `x_min+i*bin_size` and `x_min+(i+1)*bin_size` , increment `hist[i].`
    - For efficiency, try to use continue to goto the next bin and data point.
- Return `hist` and the list corresponding of the bin edges (i.e. of `x_min+i*bin_size`).