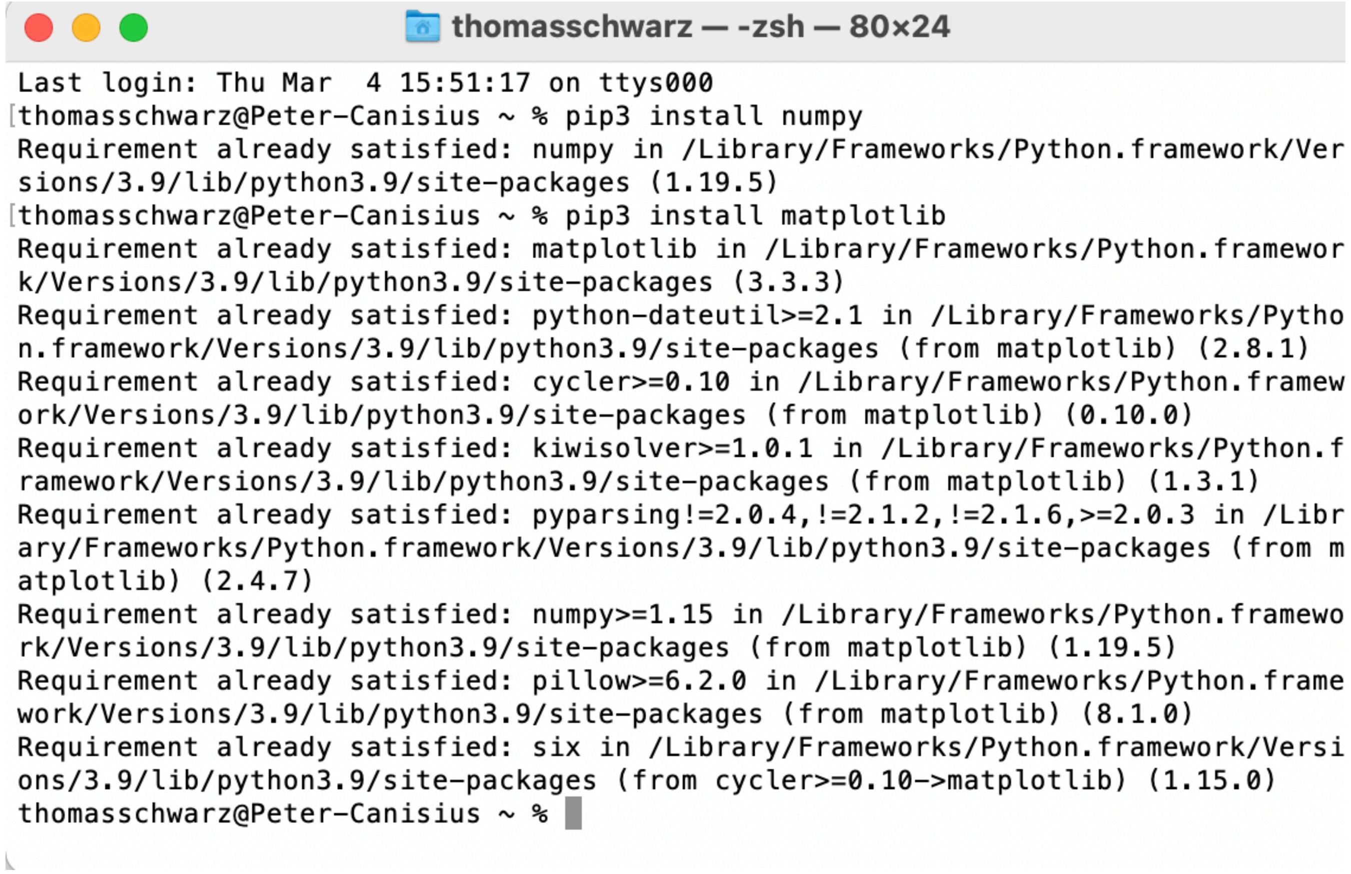
How do to counting statistics

Thomas Schwarz, SJ

Getting the Tools

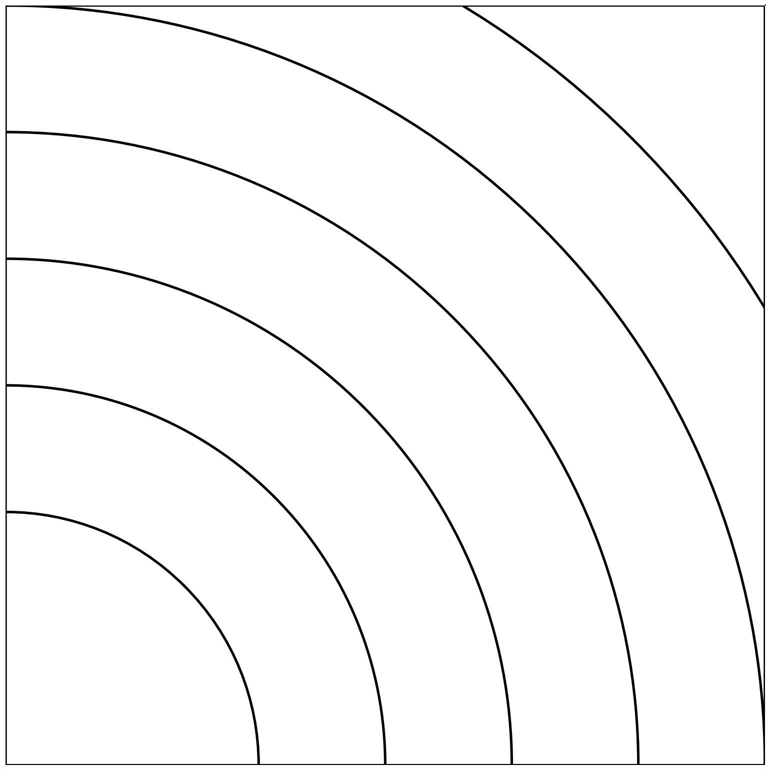
- To draw, use numpy and matplotlib
- Import the modules using pip3 from a command line
  - pip3 install numpy
  - pip3 install matplotlib
  - Takes some time



```
Last login: Thu Mar  4 15:51:17 on ttys000
[thomasschwarz@Peter-Canisius ~ % pip3 install numpy
Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/Ver
sions/3.9/lib/python3.9/site-packages (1.19.5)
[thomasschwarz@Peter-Canisius ~ % pip3 install matplotlib
Requirement already satisfied: matplotlib in /Library/Frameworks/Python.framewor
k/Versions/3.9/lib/python3.9/site-packages (3.3.3)
Requirement already satisfied: python-dateutil>=2.1 in /Library/Frameworks/Pytho
n.framework/Versions/3.9/lib/python3.9/site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: cycler>=0.10 in /Library/Frameworks/Python.framew
ork/Versions/3.9/lib/python3.9/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Library/Frameworks/Python.f
ramework/Versions/3.9/lib/python3.9/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /Libr
ary/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from m
atplotlib) (2.4.7)
Requirement already satisfied: numpy>=1.15 in /Library/Frameworks/Python.framewo
rk/Versions/3.9/lib/python3.9/site-packages (from matplotlib) (1.19.5)
Requirement already satisfied: pillow>=6.2.0 in /Library/Frameworks/Python.frame
work/Versions/3.9/lib/python3.9/site-packages (from matplotlib) (8.1.0)
Requirement already satisfied: six in /Library/Frameworks/Python.framework/Versi
ons/3.9/lib/python3.9/site-packages (from cycler>=0.10->matplotlib) (1.15.0)
thomasschwarz@Peter-Canisius ~ % 
```

Creating Binned Counts

- Toy problem: What is the distribution of the distance of random points in the unit square to the origin
- Count the number of random points with distance in
  - $[0,\frac{\sqrt{2}}{10}],$
  - $(\frac{\sqrt{2}}{10},2\frac{\sqrt{2}}{10}],$
  - $(2\frac{\sqrt{2}}{10},3\frac{\sqrt{2}}{10}],$
  - ...



Creating Binned Counts

- How do we do this:
  - First we create the bins

```
def generate_samples(nr = 100000, nr_bins=10):
    bins = [0 for i in range(nr_bins)]


   return bins
```

Creating Binned Counts

- Then we generate a large number of sample points uniformly distributed in $[0,1] \times [0,1]$

```
def generate_samples(nr = 100000, nr_bins=10):
    bins = [0 for i in range(nr_bins)]
    for _ in range(nr):
        x, y = random.random(), random.random()


    return bins
```

Creating Binned Counts

- We calculate the distance from the origin
-

```
def nrm(x,y):
    return math.sqrt(x**2+y**2)
```

Creating Binned Counts

- We know that the distance between the upper right corner and the origin is $\sqrt{2}$
- We divide the distance of a point with $n\sqrt{2}$ where n is the number of bins and round down to the nearest integer

```
def generate_samples(nr = 100000, nr_bins=10):
    bins = [0 for i in range(nr_bins)]
    for _ in range(nr):
        x, y = random.random(), random.random()
        my_bin = int(nr_bins*nrm(x,y)/math.sqrt(2))
        bins[my_bin] += 1
    return bins
```

Creating Binned Counts

- This will be the number of the bin.
- We now increment the count of the elements in the bin

```
def generate_samples(nr = 100000, nr_bins=10):
    bins = [0 for i in range(nr_bins)]
    for _ in range(nr):
        x, y = random.random(), random.random()
        my_bin = int(nr_bins*nrm(x,y)/math.sqrt(2))
        bins[my_bin] += 1
    return bins
```

Displaying as a Histogram

- To display, use matplotlib.pyplot

```
import numpy as np
import matplotlib.pyplot as plt
```

- This gives a Matlab like graphing environment

Displaying as a Histogram

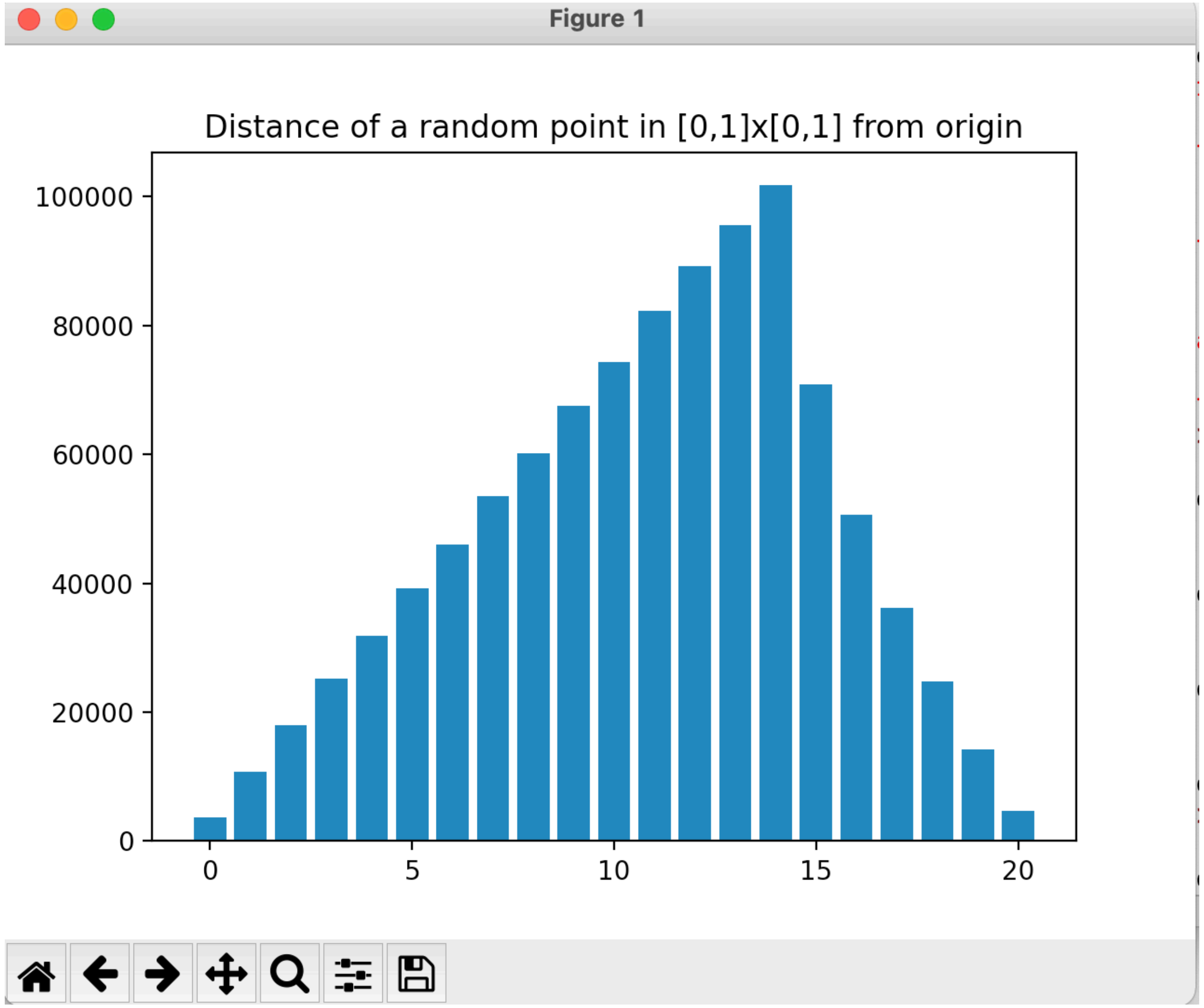
- We use the bar function
- Needs X-values, here a range
- And Y-values, here the numbers in the vector bins
- Can set a title and lots of other things

```
def display(bins):
    plt.bar( np.arange(len(bins)), bins)
    plt.title('Distance of a random point in [0,1]x[0,1] from origin')
    plt.show()
```

- Don't forget the magic incantation "plt.show( )" or nothing shows up

Displaying as a Histogram

- Result can be saved in any format



Programming Assignment

- You need to implement LH-addressing
- Instead of inserting a record into a bucket, you just increment a bin
    - So that you are just counting the number of records
- Then do everything as in the example