ASTC02 - PROF. HANNO REIN

CENTRAL LIMIT THEOREM

REMAINING LECTURES

- *Today*: Big observatories + central limit theorem
- Public observing
- Galactic distance scale
- Spectroscopy (indoors)

FINAL EXAM

- SW505B
- December 6th 9-11am
- No electronic devices
- Open book exam
- Bring any lecture notes, lab reports, etc
- Celestial spheres yes/no?

WHY DO WE USE NORMAL DISTRIBUTIONS SO MUCH?

In [7]:

```
def logl(par):
    a, l, h, k, ix, iy = par

    # simple prior
    if a<1 or a>4 or h<-0.5 or h>0.5 or k<-0.5 or k>0.5 or ix<-0.5 or ix>0.5 or iy<-0.5 or iy>0.5 :
        return -np.inf

    # relative particle position
    p_p = rebound.Particle(simulation=sim,a=a, l=l,h=h,k=k,ix=ix,iy=iy)
    p = p_p- sim.particles[1]
    x, y,z = p.xyz

    # Predict ra, dec, and derivatives
    ra = np.arctan2(y,x)%(np.pi*2.)
    dra = -y/(x**2+y**2) * p.vx + x/(x**2+y**2) * p.vy

    dec = np.arctan2(z,np.sqrt(x**2+y**2))
    ddec = -z/(x**2+y**2+z**2) /np.sqrt(x**2+y**2) * (p.x*p.vx + p.y*p.vy) + \
           np.sqrt(x**2+y**2)/(x**2+y**2+z**2) * p.vz

    # return log likelihood
    l = 0
    l += (ra-ra1)**2 / 0.00001**2  # rough estimates of accuracy (you should improve this!)
    l += (dec-dec1)**2 / 0.00001**2
    l += (dra-dra1)**2 / (0.00001 * 24 * 365.25 / (np.pi*2))**2
    l += (ddec-ddec1)**2 / (0.00001 * 24 * 365.25 / (np.pi*2))**2
    return -l
```

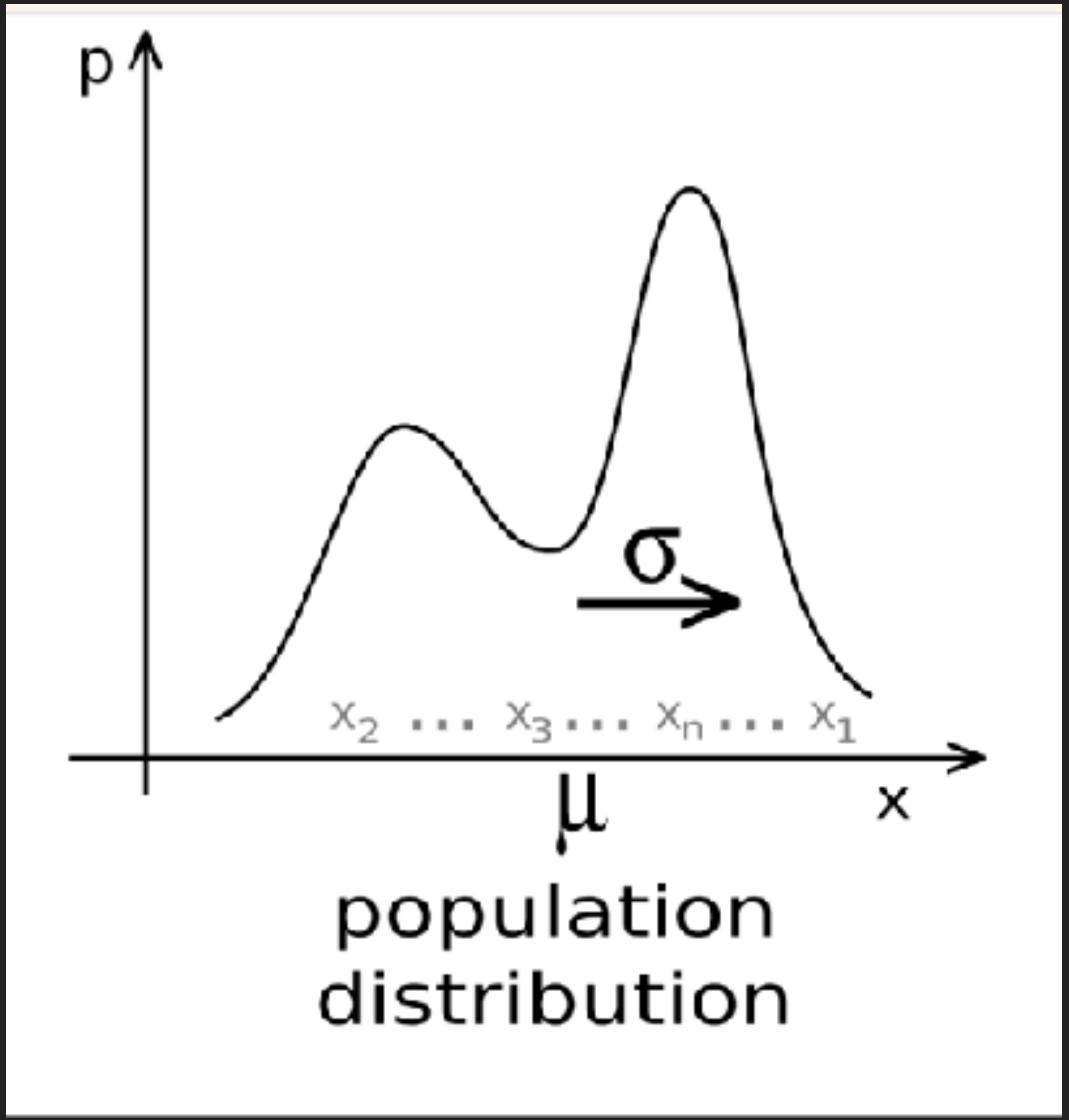
POPCORN

- 14% water
- Heating above boiling point
- Trapped inside shell
- Released if shell cracks
- Molten starch solidifies

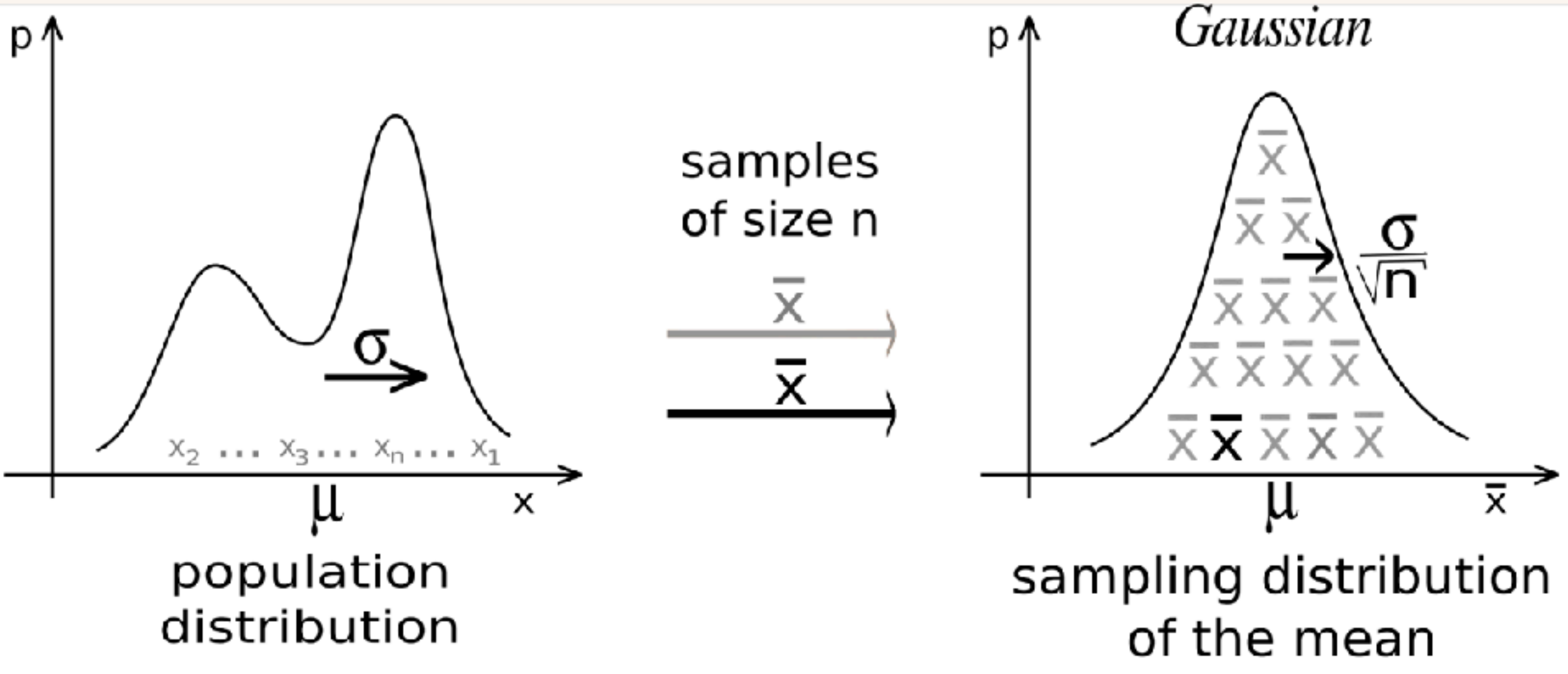
EXPERIMENT

- Heat popcorn in microwave
- Measure times of pops
- Repeat experiment many times

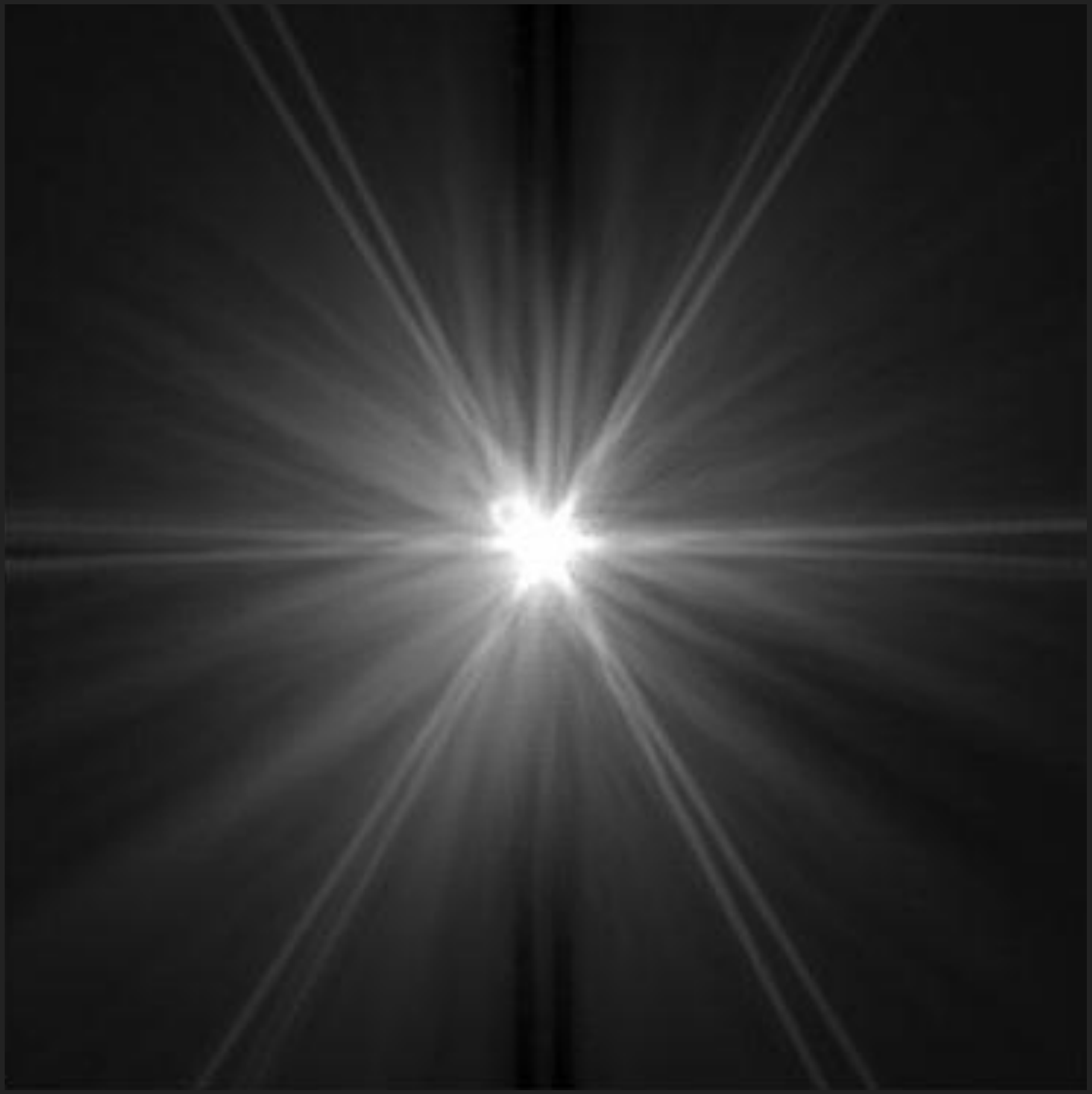
- Each kernel pops at a random time
- Distribution might be complicated
- For example: mix of small and large sizes
- Not necessarily gaussian



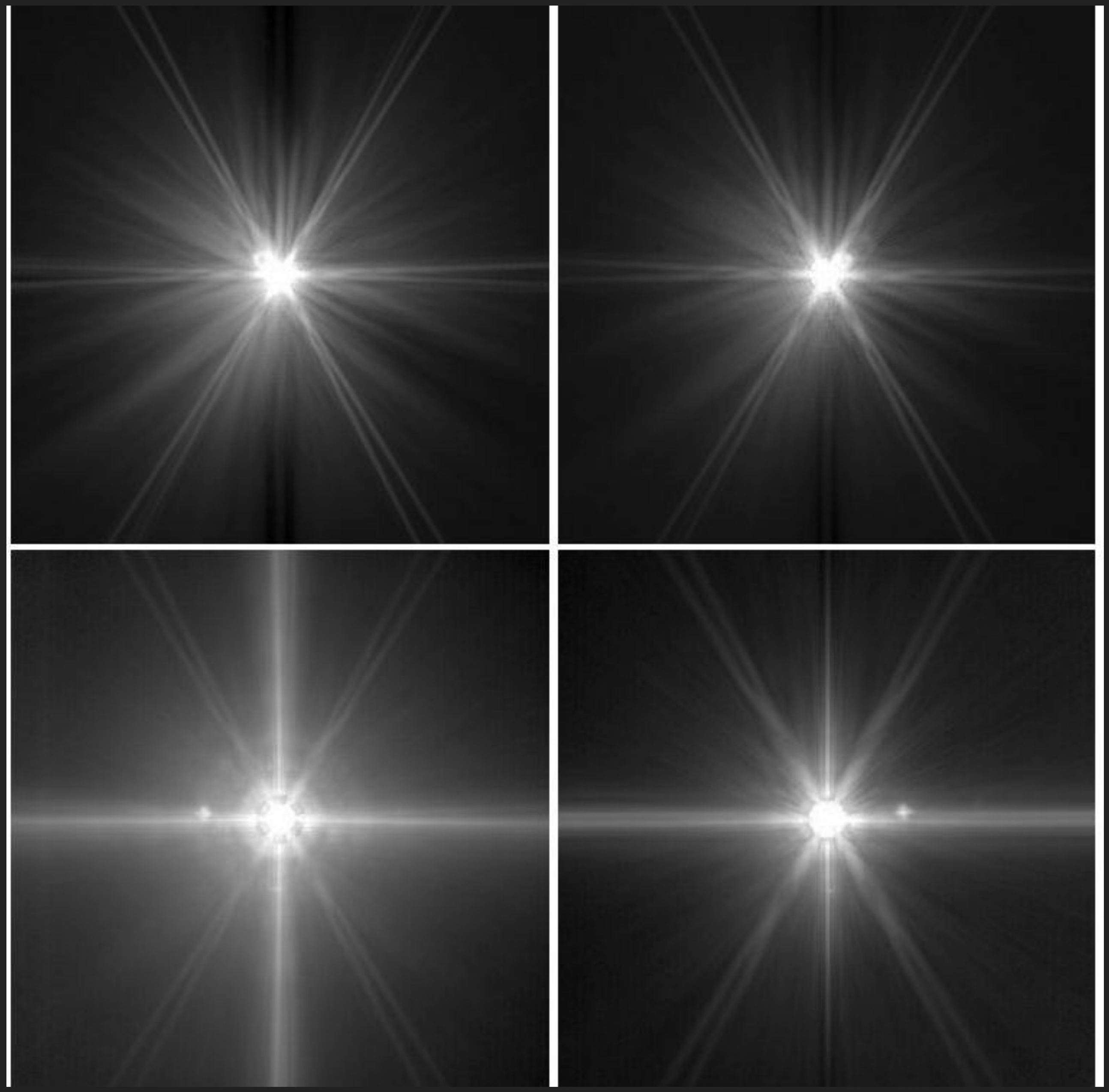
CENTRAL LIMIT THEOREM



- Each photons falls onto a random pixel
- Distribution might be complicated
- Photon position on detector does not follow a Gaussian



- But: Let's determine the mean x and y values for the star many times
- Distribution of mean x and ys will be approximately Gaussian



WHY DO WE USE NORMAL DISTRIBUTIONS SO MUCH?

In [7]:
```
def logl(par):
    a, l, h, k, ix, iy = par

    # simple prior
    if a<1 or a>4 or h<-0.5 or h>0.5 or k<-0.5 or k>0.5 or ix<-0.5 or ix>0.5 or iy<-0.5 or iy>0.5 :
        return -np.inf

    # relative particle position
    p_p = rebound.Particle(simulation=sim,a=a, l=l,h=h,k=k,ix=ix,iy=iy)
    p = p_p- sim.particles[1]
    x, y,z = p.xyz

    # Predict ra, dec, and derivatives
    ra = np.arctan2(y,x)%(np.pi*2.)
    dra = -y/(x**2+y**2) * p.vx + x/(x**2+y**2) * p.vy

    dec = np.arctan2(z,np.sqrt(x**2+y**2))
    ddec = -z/(x**2+y**2+z**2) /np.sqrt(x**2+y**2) * (p.x*p.vx + p.y*p.vy) + \
           np.sqrt(x**2+y**2)/(x**2+y**2+z**2) * p.vz

    # return log likelihood
    l = 0
    l += (ra-ra1)**2 / 0.00001**2  # rough estimates of accuracy (you should improve this!)
    l += (dec-dec1)**2 / 0.00001**2
    l += (dra-dra1)**2 / (0.00001 * 24 * 365.25 / (np.pi*2))**2
    l += (ddec-ddec1)**2 / (0.00001 * 24 * 365.25 / (np.pi*2))**2
    return -l
```