Social Data Science: Machine Learning & Econometrics

Exercise class 9

May 11, 2020

Todays quick warmup

Consider a sequence $\{(x_k, y_k)\}$ of points in $\mathbb{R}^2$ where

- $\forall k : x_{k+1} > x_k$ and $y_{k+1} > y_k$
- $\forall k : x_k \in [0, 1]$
- $\forall k : y_k \in [0, 1]$

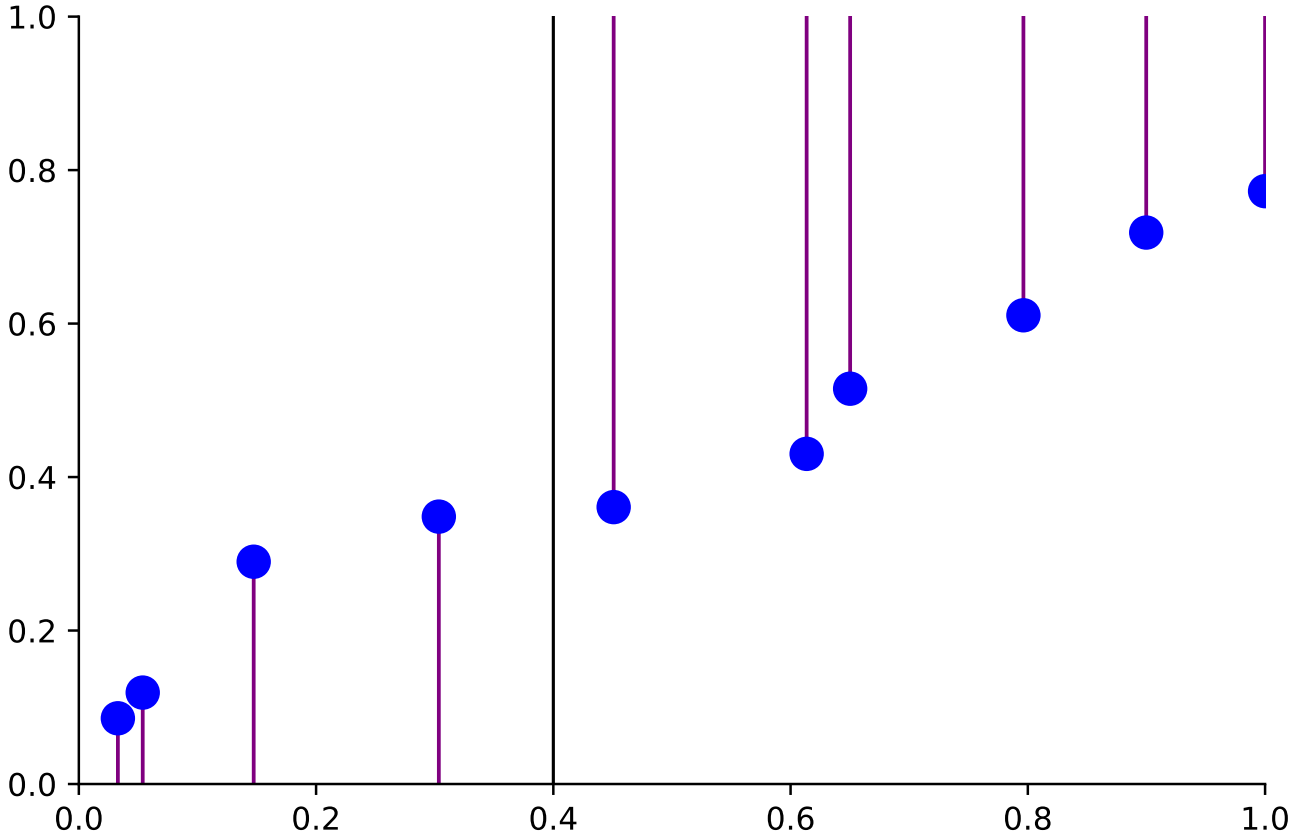
Q1: Write a function `simulate(n)` that simulates such a sequence with length n by drawing x and y according to $z_{k+1} - z_k \sim U(0, 2/n)$ and clipping any $z_k > 1$ to the bounding box.

Todays quick warmup

Q2: By the *divided sum at* x_0 of such a sequence we mean the the sum

$$s(x_0) = \sum_k y_k \mathbf{1}_{(x_k < x_0)} + (1 - y_k)\mathbf{1}_{(x_k \geq x_0)} \tag{1}$$

I.e. the sum of the vertical bars marked in the figure below. Write a function `minimize` that takes as its inputs a simulated sequence and returns x_k^* that minimizes $s(x_0 = x_k)$ as well as the corresponding y_k^*.



Todays quick warmup

Q3: Run simulate with $n =$`np.arange(10,1000,10)` and plot the resulting y_k^* against n. What value does y_k^* seem to converge to? Does this make sense to you?

Todays quick warmup - solution

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

def simulate(n):
    x0 = np.random.uniform(0, 2/n)
    y0 = np.random.uniform(0, 2/n)
    x, y = [x0], [y0]
    for _ in range(n-1):
        x0 = x0 + np.random.uniform(0,2/n)
        y0 = y0 + np.random.uniform(0,2/n)
        x.append(x0 if x0 < 1 else 1)
        y.append(y0 if y0 < 1 else 1)
    return x, y
```

Todays quick warmup - solution

```
def minimize(x,y):
    x0_star = 0
    y0_star = 1
    s_star = sum([1- yk for yk in y])

    for x0,_ in zip(x,y):
        s1 = sum([yk for xk, yk in zip(x,y) if xk < x0])
        s2 = sum([1 - yk for xk, yk in zip(x,y) if xk >= x0])

        if s1 + s2 < s_star:
            s_star = s1 + s2
            x0_star = x0
            y0_star = max([yk for xk, yk in zip(x,y) if xk < x0]

    return x0_star, y0_star
```