

```
In [ ]: #3a + 3b
def stdev(nums: list[float]) -> float:
    mean = sum(nums) / len(nums)
    variance = sum([(num - mean)**2 for num in nums]) / len(nums)
    return variance**(1/2)

print(stdev([3,1,4,1,5,9,2,6,5,3]))

2.3430749027719964
```

```
In [ ]: #3.c
def stdev2(nums: list[float], sample=False) -> float:
    if len(nums) <= 1 - (not sample):
        raise RuntimeError("Not enough numbers provided")
    mean = sum(nums) / len(nums)
    variance = sum([(num - mean)**2 for num in nums]) / (len(nums) - 1 *
    return variance**(1/2)
```

```
In [ ]: #3d
print(stdev2([3,1,4,1,5,9,2,6,5,3], True))

2.4698178070456938
```

```
In [ ]: #3e
try:
    print(stdev2([]))
except RuntimeError as e:
    print(e)
try:
    print(stdev2([], True))
except RuntimeError as e:
    print(e)
try:
    print(stdev2([3]))
except RuntimeError as e:
    print(e)
try:
    print(stdev2([3], True))
except RuntimeError as e:
    print(e)
```

Not enough numbers provided  
 Not enough numbers provided  
 0.0  
 Not enough numbers provided

```
In [ ]: #4e
import math
def closest(nums: list[float], val: float):
    if len(nums) == 0:
        return -1
    min = abs(nums[0] - val)
    min_index = 0
    for i, num in enumerate(nums):
        if abs(num - val) < min:
            min = num - val
            min_index = i
    return min_index
```

```
print(closest([1,1,2,3,5,8,13,21], math.pi))
```

2