ex.) generate 5,000 random #s from 1 to 100 and calculate how many are even

```
empty array
for i in range(5000)
    generate random #
    store it in array (np.append)
```

→ array with 5000 values
check how many are even

---

another way - uses for loop but no array

define counter = 0
for i in range(5000):
    generate random #
    if it's even:
        add 1 to counter

accumulator pattern

---

most simulations use array to store value (like test statistic)

---

df of students

name	college	age
Joe Schmoe	Sixth	20
Jane Smith	Roselle	15

how many students have last name that starts with S

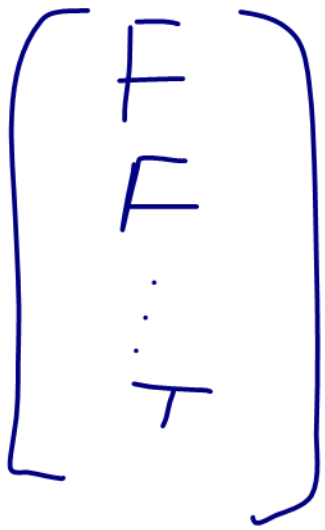
```
def last_name_S (one_name):
    return one_name.split(" ")[1].
        startswith("S")
```

```
Students.get("name").
apply(last_name_S).
sum()
```

T=1, F=0

Series string methods work on entire column directly

students.get("name")
.str.contains("Eli")

⇒ 

← Elizabeth Jones
or Jeff Elixir

Key idea:

.str before a string method does the string method on the entire series

name	college	age
Joe Schmoe	Sixth	20
Jane Smith	Revelle	15

how many students have last name that starts with S

```
[students.get("name").str.split(" ").str[1].str.startswith('S')
 [["Joe","Schmoe"],["Jane","Smith"]]
```

clever way

students.get("name").
str.contains(" S")

space

---

bootstrapping

1) resample size = original sample size
2) with replacement

problem 4 from FA24 final

4.1

86% CI

$\Rightarrow 100 - 86 = 14\%$ chopped

$\frac{14}{2} = 7\%$ off each side

$7, \quad 100 - 7 = 93$

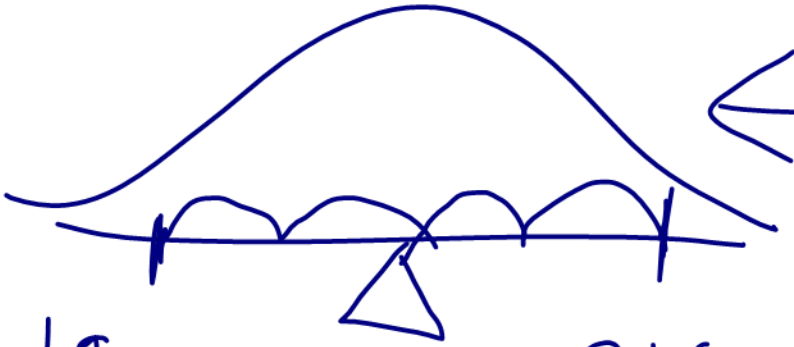
4.2

95% CLT CI $\Rightarrow [9.54, 21.58]$

ask for SD of pop

SD of dist of sample means $= \frac{\text{pop SD}}{\sqrt{400}}$

pop SD = 20 × SD of sample means



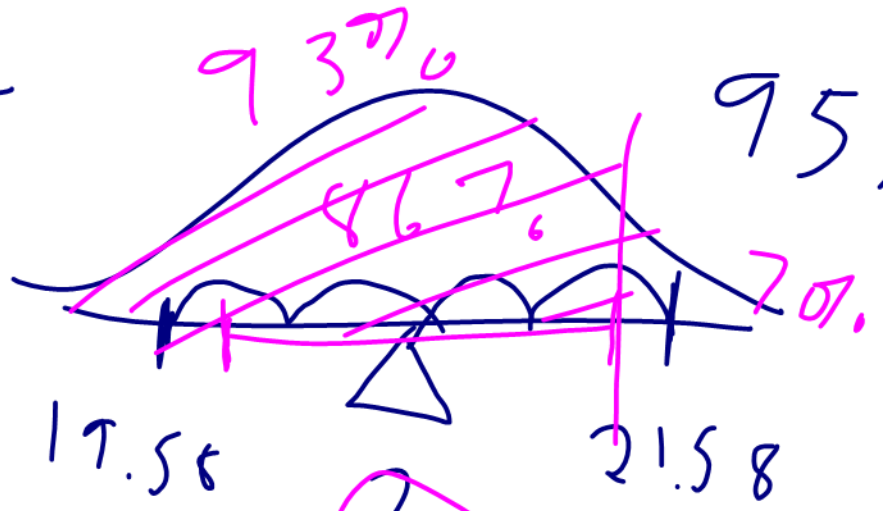
dist of sample means

19.5 20.5 21.5

one "jump" = one SD of dist of sample means = 0.5

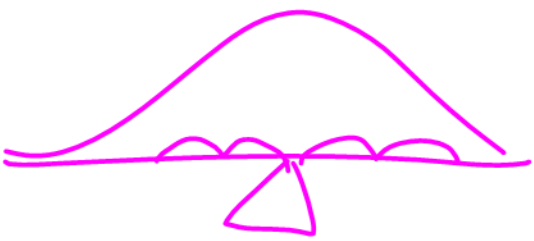
pop SD = 20 × 0.5 = 10

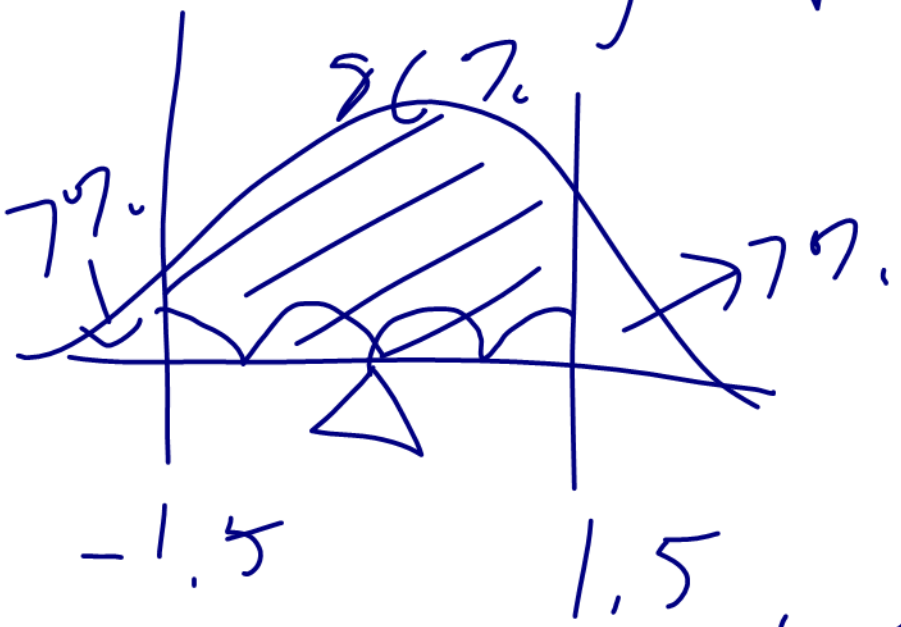
4.3



95% CLT

20.58





endpoints of 86% CI

$20.58 \pm 1.5 * 0.5$

$$\frac{3}{2} \times \frac{1}{2}$$

$$0.75$$