

# **internshipPlacementSelectionSystem**

## **For RxCU85**

**Suparvit P.**



[github.com/iamsuparvit](https://github.com/iamsuparvit)  
iamsuparvit



/internshipplacementselectionsystem

### internshipPlacementSelectionSystem

A Jupyter Notebook for assigning students to internship sites based on ranked preferences, seat capacity, sex requirements, and two separate shifts, with built-in lottery handling and verification ...

● Jupyter Notebook

 iamsuparvit	Update the output file for exporting	56ec7e8 · 4 minutes ago	⌚ 23 Commits
 LICENSE	Add license description	2 hours ago	
 README.md	Add branch information to output file description	6 minutes ago	
 drugstore_path.csv	Add program file and example csv files	3 months ago	
 internshipPlacementSelectionSystem.ipynb	Update the output file for exporting	4 minutes ago	
 student_selection.csv	Add sample student_selection file	3 hours ago	
 test_cleanDataDrugstores67.ipynb	Add test_cleanDataDrugstores67.ipynb file	2 hours ago	

# internshipPlacementSelectionSystem

Annually, pharmacy students face the complex task of assigning every student to a limited number of internship sites, each with unique requirements. This process must be efficient, equitable, and transparent.

The project automates assigning pharmacy students at Chulalongkorn University to internships at drugstores and hospitals. Its objective is to ensure a fair, efficient, and fully auditable placement for every student that respects all rules.

**Easily Configurable**

**Rank-choice Allocation**

**Constraint Enforcement**

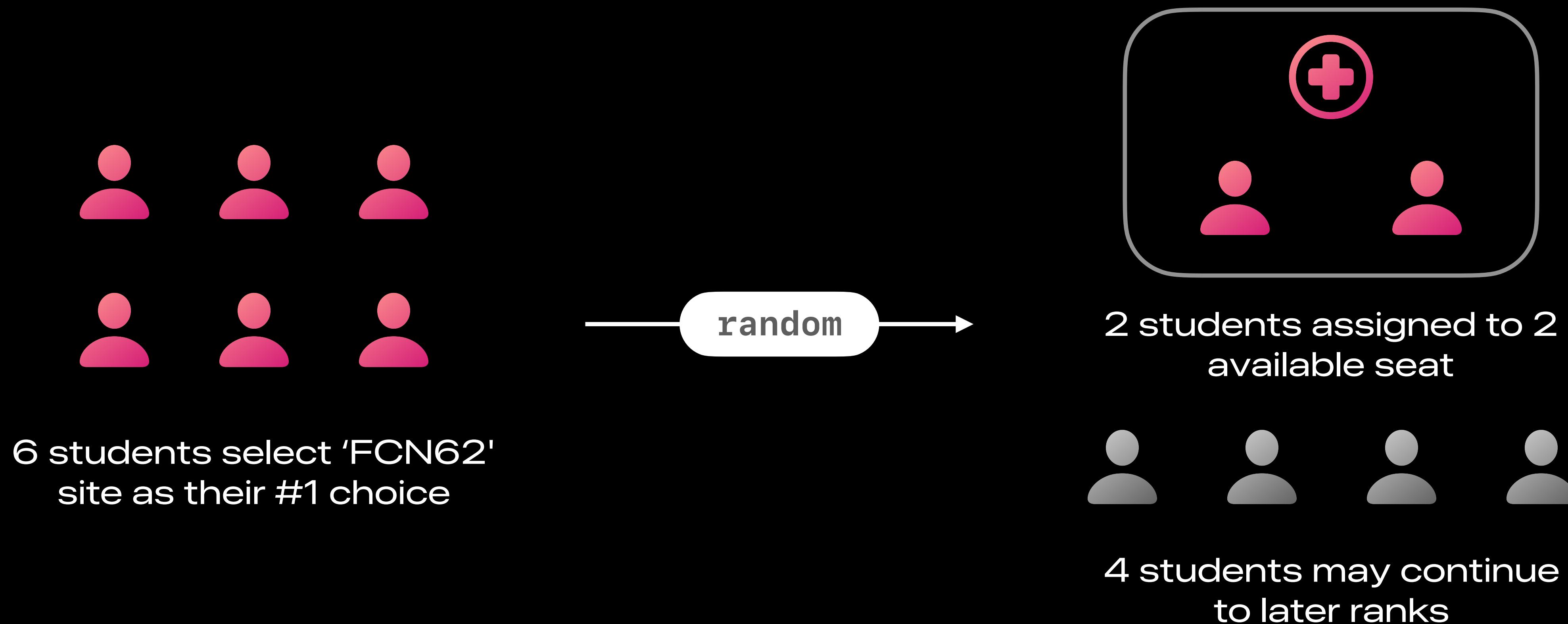
**Comprehensive Verification**

**Insightful Analysis**

**Precisely Repeatable**

# Impartial Tie-breaker

If multiple students are tied (i.e., they chose the same site at the same rank) and there are not enough seats for all, a random lottery is conducted only among that specific group. This is the fairest and most defensible method to resolve ties.



# Input / student\_df

Unique identifier

Preference orders

	student_name	student_id	sex	shift	rank1	rank2	rank3	rank4	rank5
0	d4743cd5a0e8a401b83d139eb924235143cf5394a28730...	6630002133	male	2	FCN49	FCN49	FCN49	FCN49	FCN49
1	ee5fa7606916cc9994d1c1a6f986f3b4926377c49d8923...	6630003833	male	2	FCN62	FCN60	FCN61	FCN64	FCN59
2	79cb28ef79b086da8546693c0c362f957d3211d403fa2b...	6630004433	male	1	BTS05	HPT01	BTS20	BTS14	BTS12
3	6e5cf96bde58559be4f0423fa9c3149eae506f78c3b862...	6630006733	female	1	FCN68	FCN55	PUR04	FCN67	FCN66
4	e50488993ca3e7c428c8d21ccc546a588093234e6fb6d2...	6630008033	female	2	FCN41	FCN32	CEN03	BTS15	FCN31
...	...	...	...	...	...	...	...	...	...
129	a7cad6d2921e9e434767c2fb4179c936ff2b8bad41bb3f...	6631086033	female	2	FCN32	FCN31	BTS15	FCN41	CEN03
130	fd82a210bcc52c26ba4fdbfb3794c35aea81cb4768b15e...	6631087733	female	2	BTS10	FCN14	BKK08	BKK09	BKK26
131	b792349f7482cdb30e7f7958829c952817420a4cecbffa...	6631088333	female	2	FCN76	FCN05	FCN73	FCN80	BTS06
132	dfb13aaafdf6ff9e61c8ed90142c994ec9c3cf649266fc4...	6631090533	male	1	BKK31	BKK30	BKK44	BKK36	BTS09
133	92c187ee3fbfbf5800c9c898553c38ba23f67da02c912f...	6631092833	male	1	BTS05	BTS03	BTS16	CEN05	BTS12

The student's name has been hashed for privacy reasons

# Input / drugstore\_df

Sex requirement (male, female, or both)

Maximum capacity

Unique identifier

	code	branch	sex_require1	seat1	sex_require2	seat2
0	CEN01	6 ஸ்ரோவ்னாரமஹங்	both	2.0	both	2.0
1	CEN02	7 முழுமீ பூருாசர்ங்ஸர்க்	both	2.0	NaN	NaN
2	CEN03	19 வங்கீஸ்வாங்	male	1.0	male	1.0
3	CEN04	20 பொம்பிராப்க்ட்ரூப்யை	NaN	NaN	both	2.0
4	CEN05	23 லீப்ரயா	male	1.0	male	1.0
...	...	...	...	...	...	...
240	FCN93	காலதீக்சீ ச்ஜூரி-ரங்ஸிட் (கலங்க)	both	2.0	both	2.0
241	FCN94	டேஓப்ரோம் தின்டெங்	both	1.0	both	1.0
242	FCN95	பீக்சீ கிரீன்வெல்ட் ப்ரஹாம் 2	both	1.0	both	1.0
243	FCN96	கிரிராச்	both	2.0	both	2.0
244	FCN97	பத்யா	both	4.0	both	4.0

# Workflow

## Configuration & Import

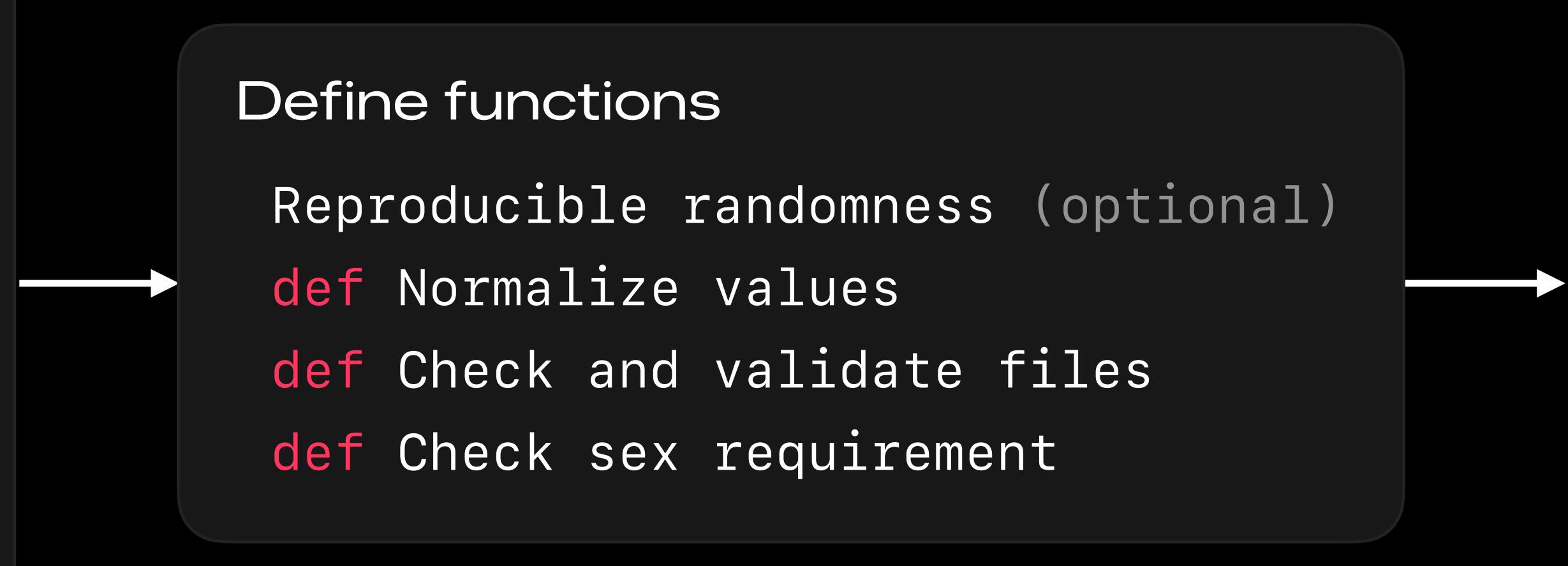
```
x (rank) = 5  
SHIFT_ORDER = 2  
student_path = "student_selection.csv"  
drugstore_path = "drugstore_path.csv"  
output_path = "./output"  
random_seed = None
```

## Define functions

```
Reproducible randomnes  
def Normalize values  
def Check and validate  
def Check sex requirem
```

# Workflow

```
Import  
1  
2  
 2  
  = "student_selection.csv"  
  n = "drugstore_path.csv"  
  "../output"  
None
```



Load and run

- Load DataFrame and validate
- Normalize data
- Remove duplicate ranks
- Create output table
- Run allocation loops

# Workflow

ctions  
ole randomness (optional)  
lize values  
and validate files  
sex requirement



## Load and run

Load DataFrame and validate columns  
Normalize data  
Remove duplicate ranks  
Create output table  
Run allocation loops



## Verification

Row counts match between  
No duplicate student IDs  
Rank correctness  
Invalid shifts  
Assigned sites that do not exist  
Capacity violations  
Duplicate ranks (again?)  
Sex requirement violations  
Students who chose unavailable sites  
Create `YYYYMMDD\_HHMMSS` timestamp

# Workflow

## Load and run

Load DataFrame and validate columns

Normalize data

Remove duplicate ranks

Create output table

Run allocation loops

## Pick a shift

### Pick a rank

#### Check each student

are not assigned yet

are in the current shift

chose a valid site at this rank

meet the site's sex requirement

and the site still has seats left in that shift

## Students grouped by the site

If seats are enough → everyone in that group gets assigned

If too many students want it → runs a random lottery and assigns only as many as there are seats

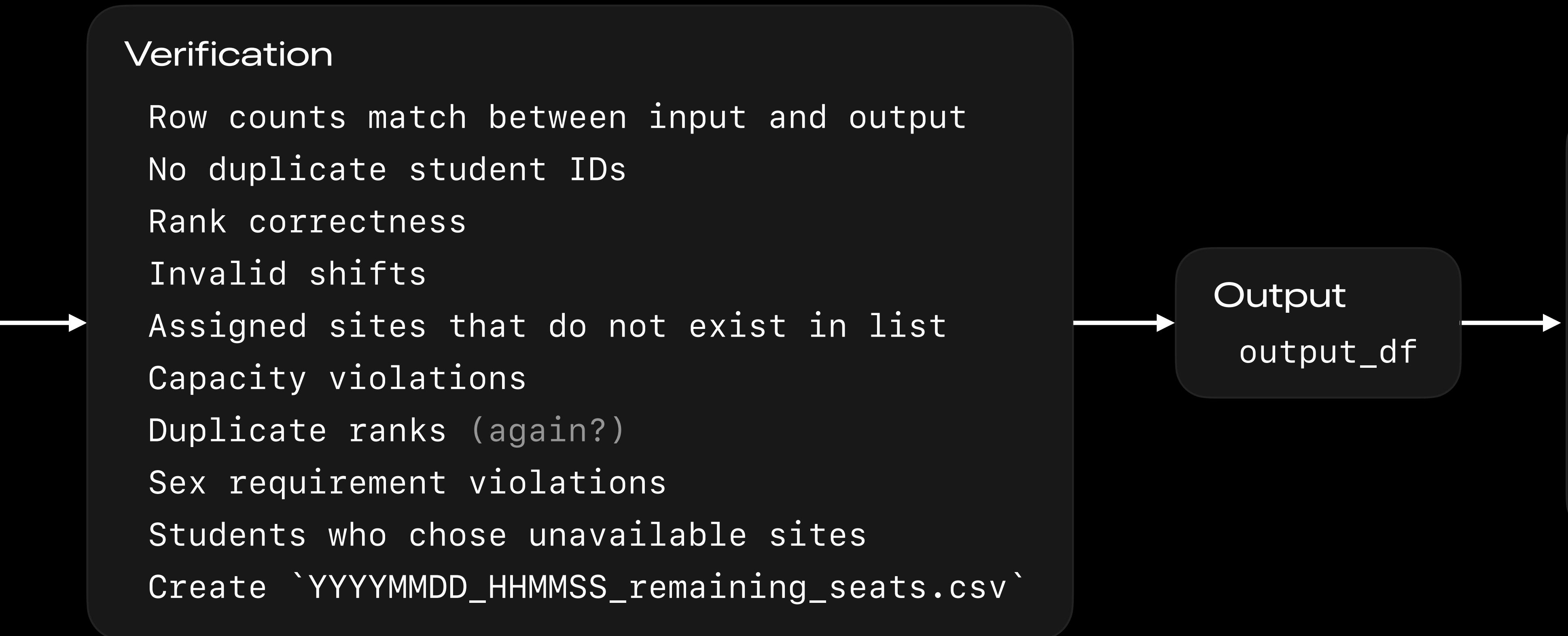
## Reduce the remaining seat count

```
for shift_num_int in range(1, SHIFT_ORDER + 1):
    for rank_num in range(1, x + 1):
        ...
        ...
```

- one shift at a time and one rank at a time
- if student doesn't meet all requirements, they remain unassigned for that rank

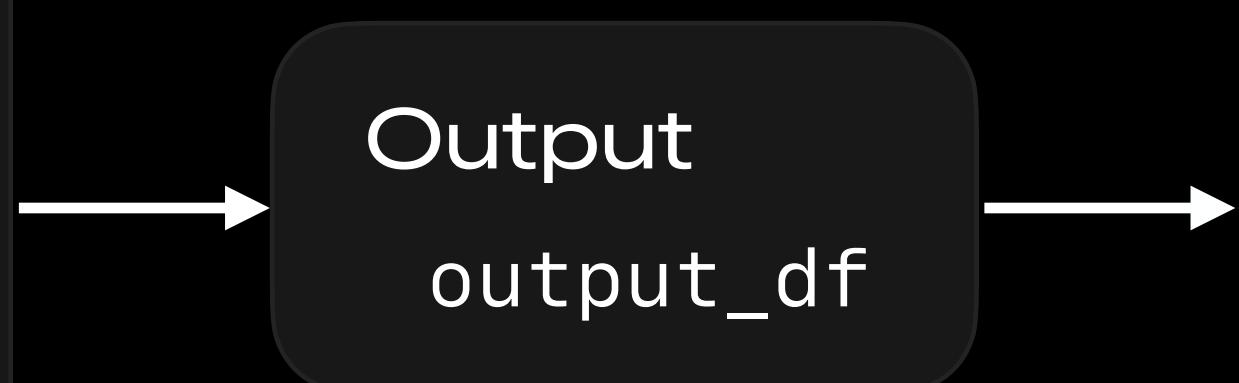
# Workflow

and validate columns  
e ranks  
able  
loops



# Workflow

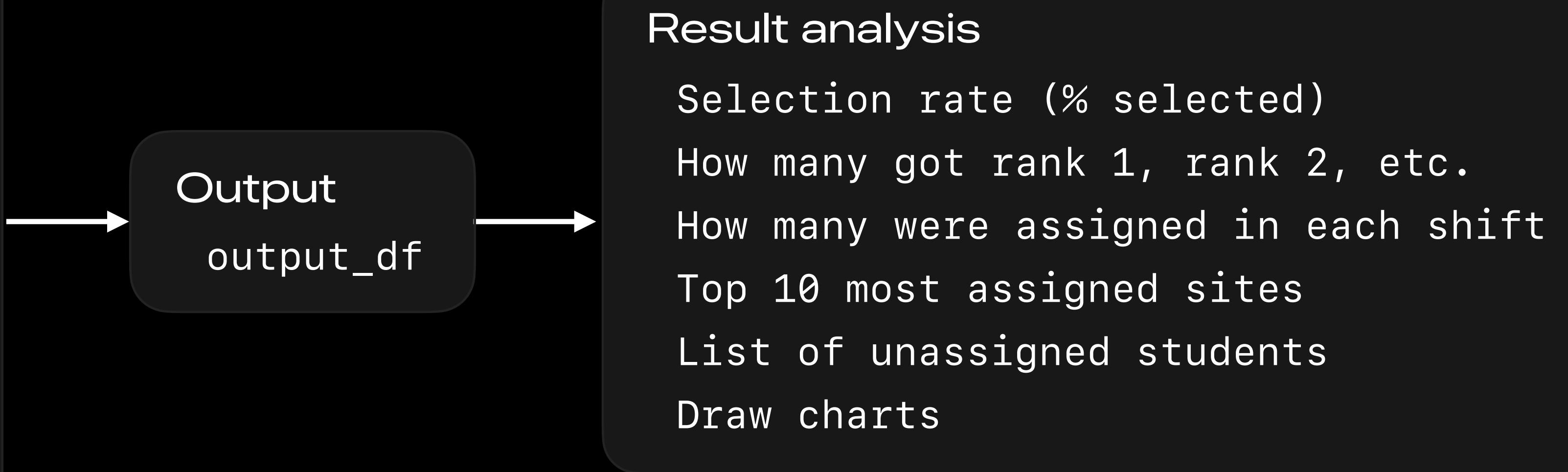
on  
nts match between input and output  
icate student IDs  
rectness  
shifts  
d sites that do not exist in list  
y violations  
te ranks (again?)  
uirement violations  
s who chose unavailable sites  
'YYYYMMDD\_HHMMSS\_remaining\_seats.csv'



## Result analysis

- Selection rate (% selected)
- How many got rank 1, rank 2, etc.
- How many were assigned in each shift
- Top 10 most assigned sites
- List of unassigned students
- Draw charts

# Workflow



# Output / output\_df

		student_name	student_id	rank_result	result	branch
0	d4743cd5a0e8a401b83d139eb924235143cf5394a28730...		6630002133	0	Not selected	
1	ee5fa7606916cc9994d1c1a6f986f3b4926377c49d8923...		6630003833	1	FCN62	บ้านดุ'
2	79cb28ef79b086da8546693c0c362f957d3211d403fa2b...		6630004433	1	BTS05	Q House Lumpini
3	6e5cf96bde58559be4f0423fa9c3149eae506f78c3b862...		6630006733	1	FCN68	ตลาดหนองดอก ลำพูน
4	e50488993ca3e7c428c8d21ccc546a588093234e6fb6d2...		6630008033	1	FCN41	คลาล เท็กซ์-ราชพฤกษ์
...		...	...	...	...	...
129	a7cad6d2921e9e434767c2fb4179c936ff2b8bad41bb3f...		6631086033	1	FCN32	งามวงศ์วาน
130	fd82a210bcc52c26ba4fdbfb3794c35aea81cb4768b15e...		6631087733	1	BTS10	The Mall Bangkapi 2
131	b792349f7482cdb30e7f7958829c952817420a4cecbffa...		6631088333	1	FCN76	ปตท. แหลมฉบัง
132	dfb13aaafdf6ff9e61c8ed90142c994ec9c3cf649266fc4...		6631090533	1	BKK31	รามอินทรา 67
133	92c187ee3fbfbf5800c9c898553c38ba23f67da02c912f...		6631092833	2	BTS03	Sathorn Square

# What if?

- ❗ A student chose the same site multiple times

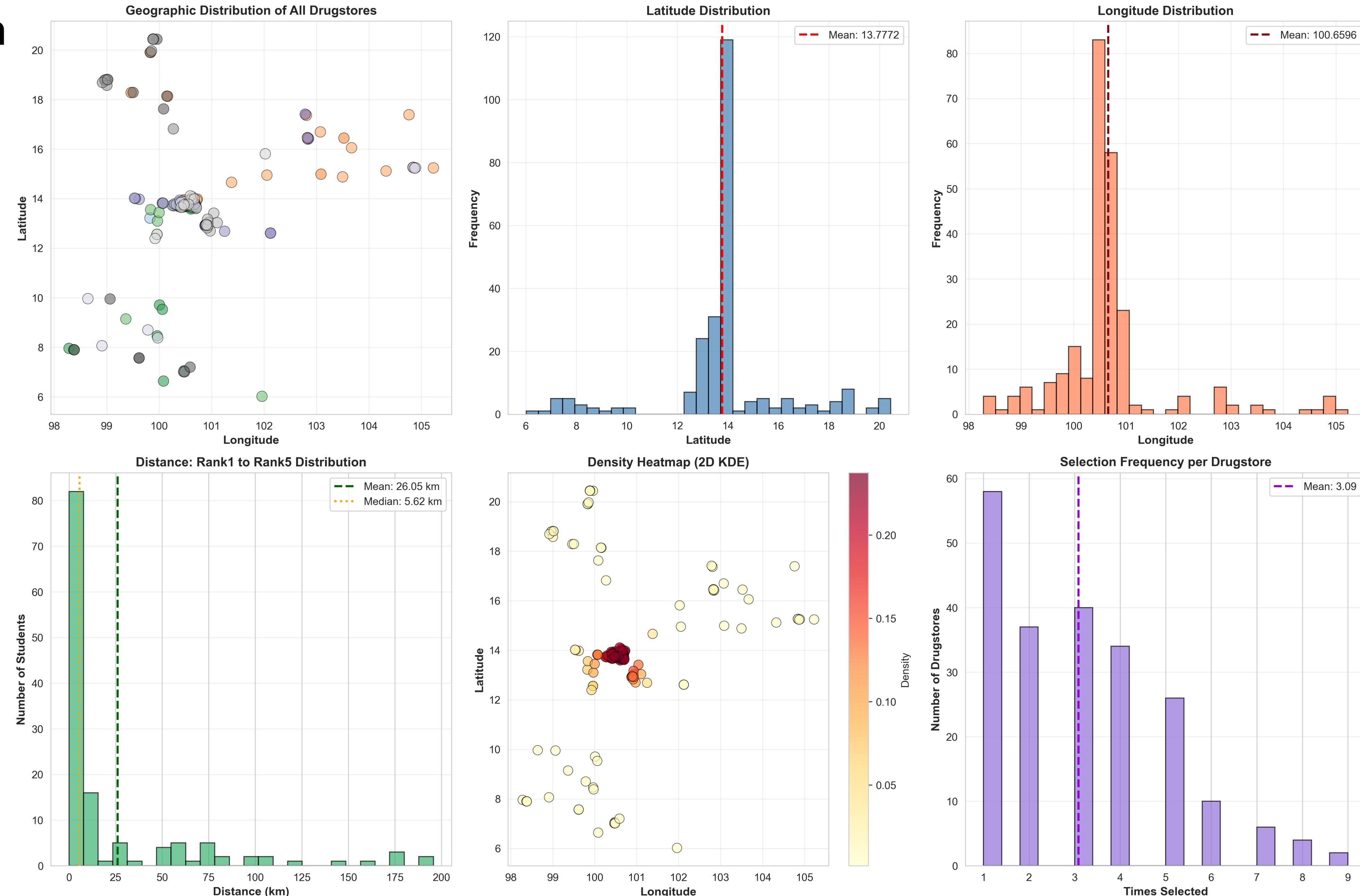
  - ✅ System blanks out the later duplicate rank cell and counts it in dup\_removed.
  - 👀 In cell 3, step 5
  
- ❗ A student chose the site that is unavailable

  - ✅ That choice is ignored, student may continue to later ranks.
  - 👀 In step 9 loop
  
- ❗ A student chose the mismatch sex-requirement site

  - ✅ That choice is ignored, student may continue to later ranks.
  - 👀 In step 9 loop
  
- ❗ A student chose choosing invalid sites (impossible if pre-option input used)

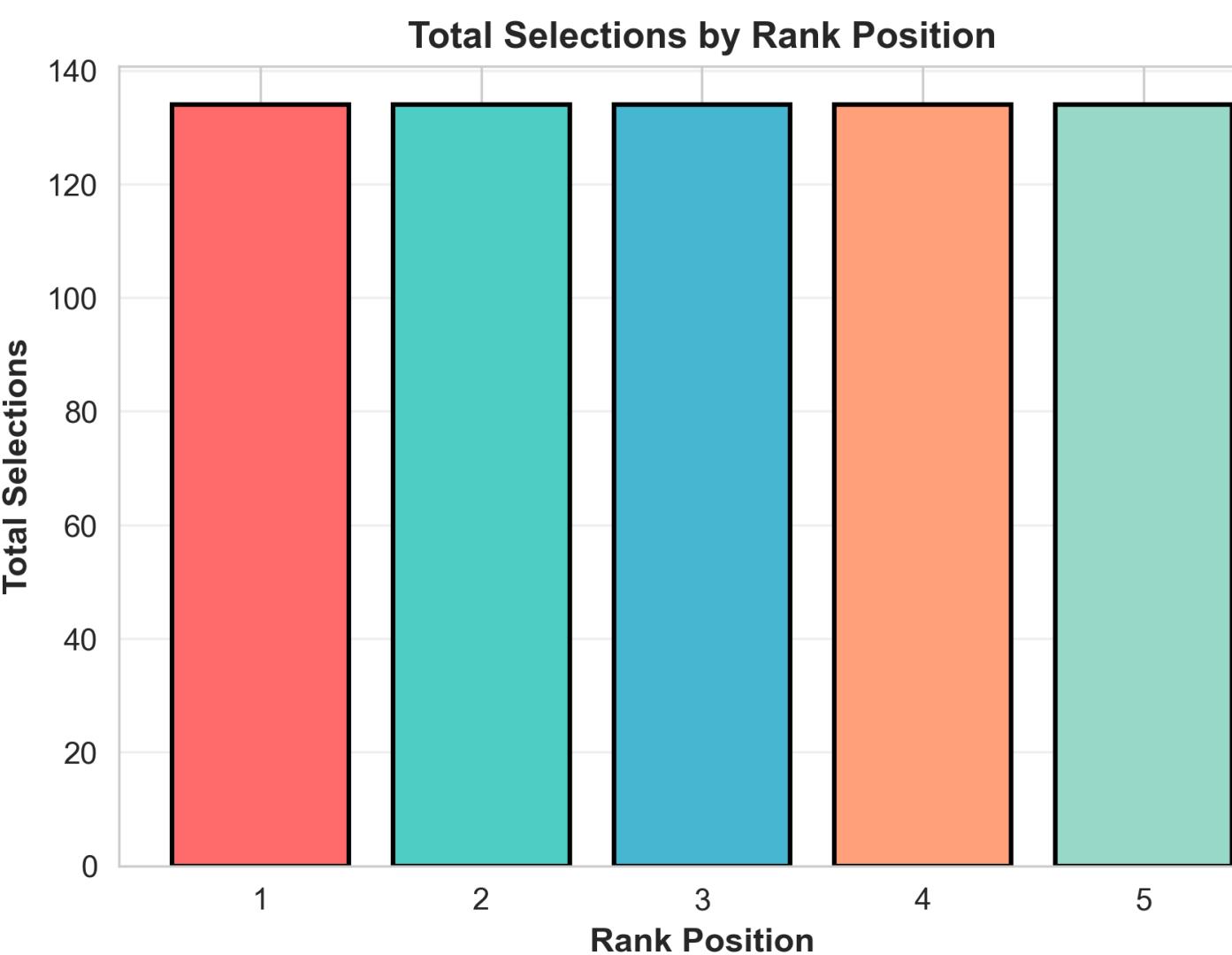
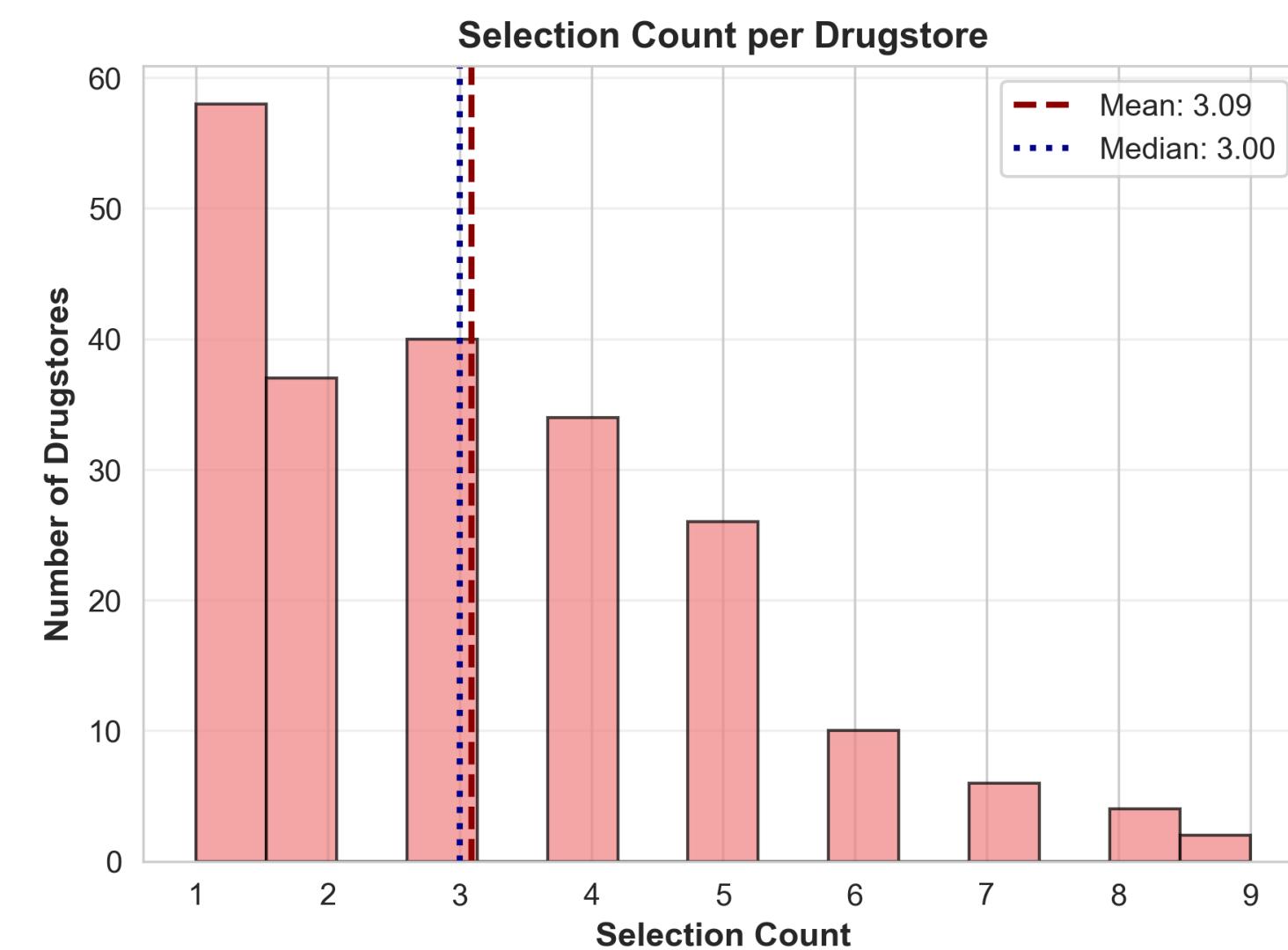
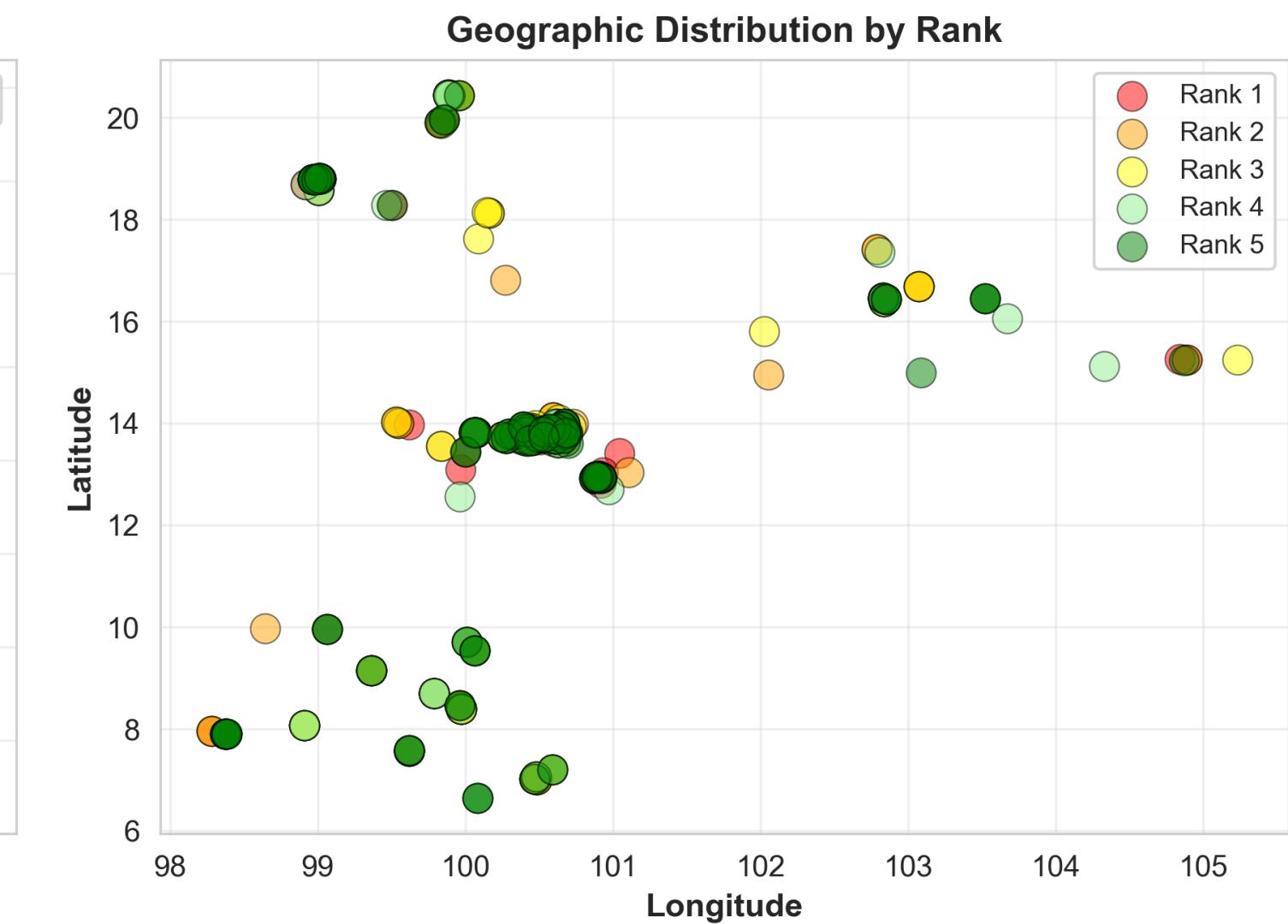
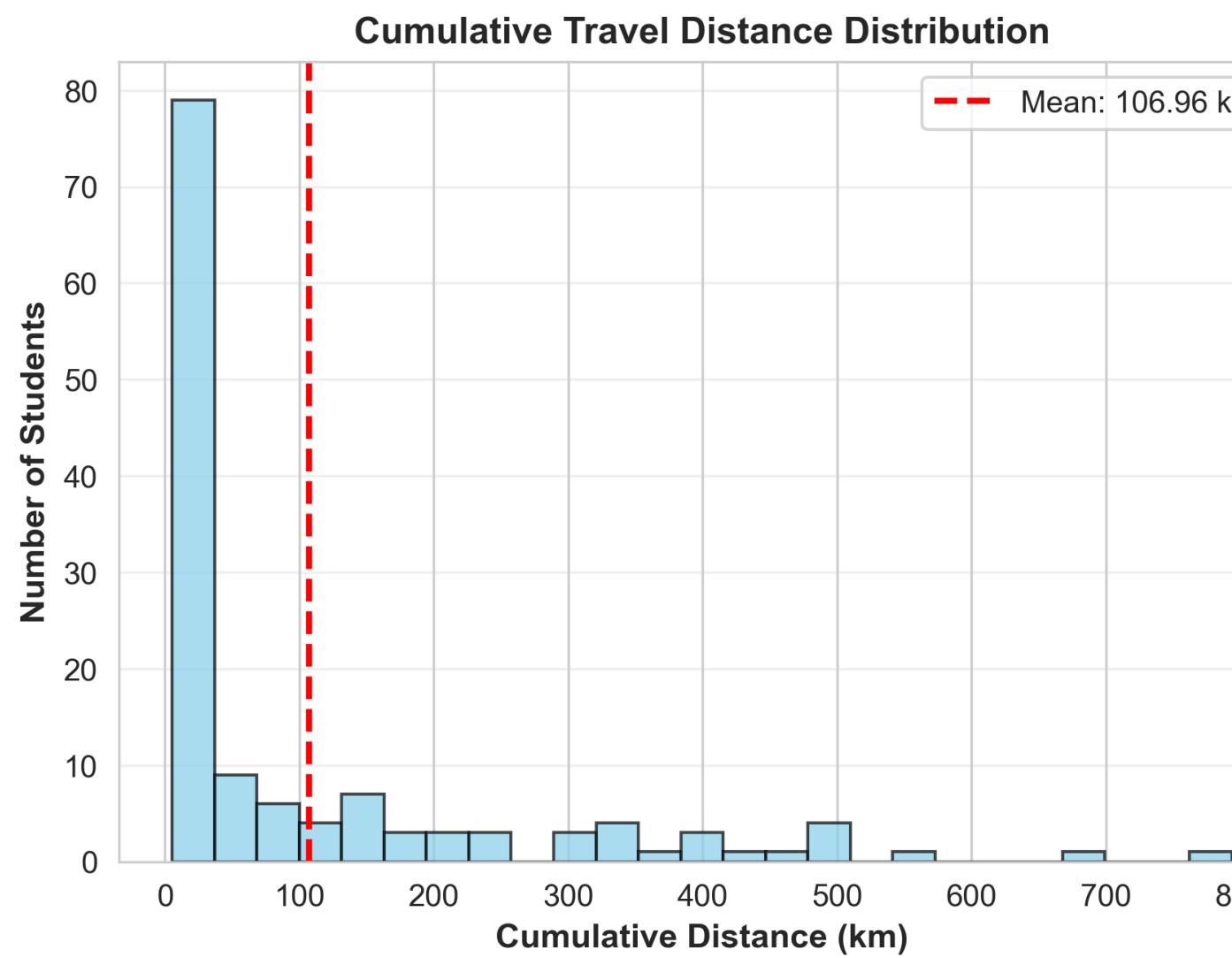
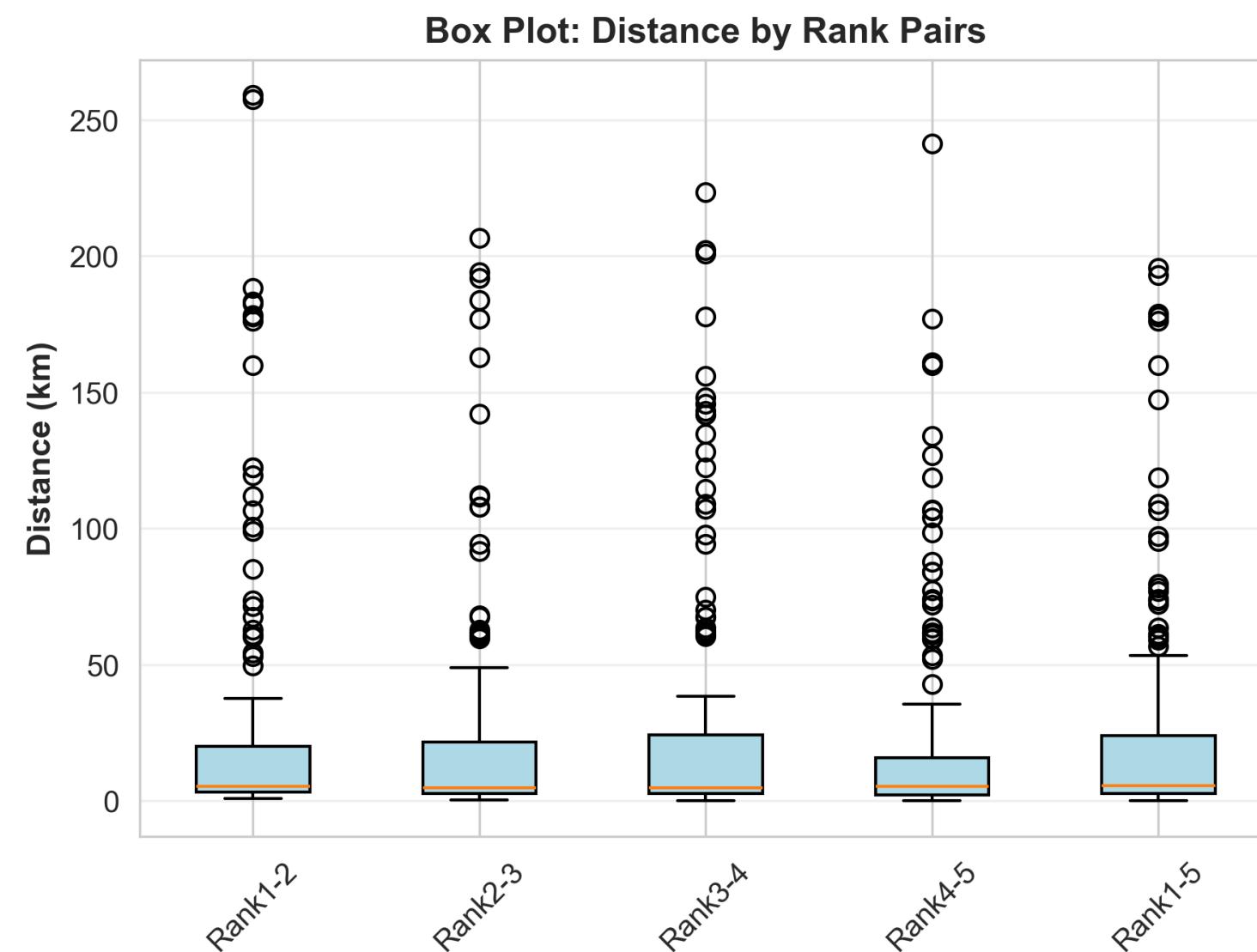
  - ✅ That choice is ignored, student may continue to later ranks.
  - 👀 In step 9 loop

# Sample Data / Analysis



The latitude and longitude of the sites are retrieved from Google Maps using the geopy Python library

# Sample Data / Analysis



**DISTANCE STATISTICS SUMMARY**

---

Total Students: 134  
Total Drugstores: 245

---

Rank1-5 Distance (km):

- Mean: 26.05
- Median: 5.62
- Min: 0.00
- Max: 195.68
- Std: 44.31

---

Cumulative Distance (km):

- Mean: 106.96
- Median: 22.54
- Min: 5.32
- Max: 793.70

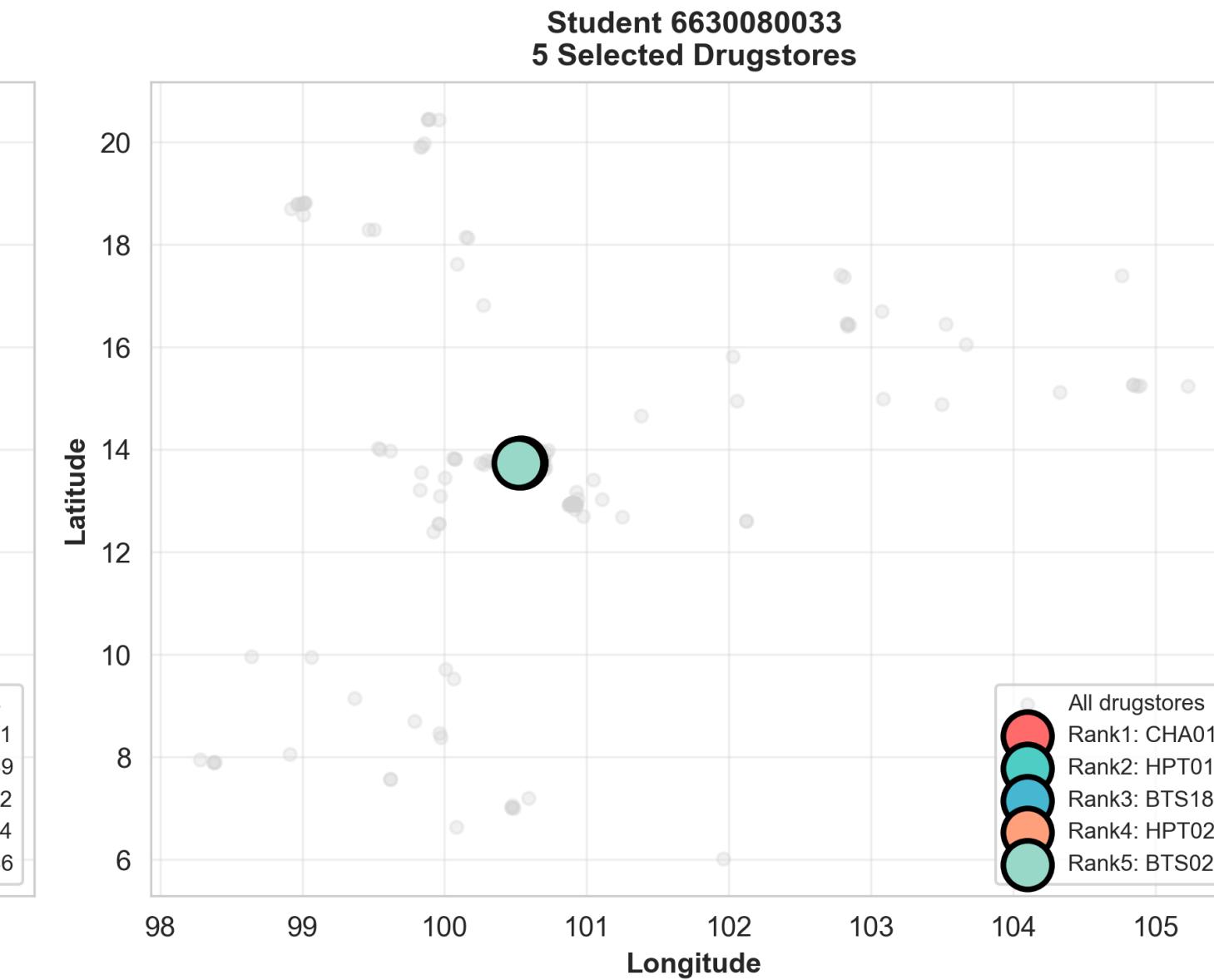
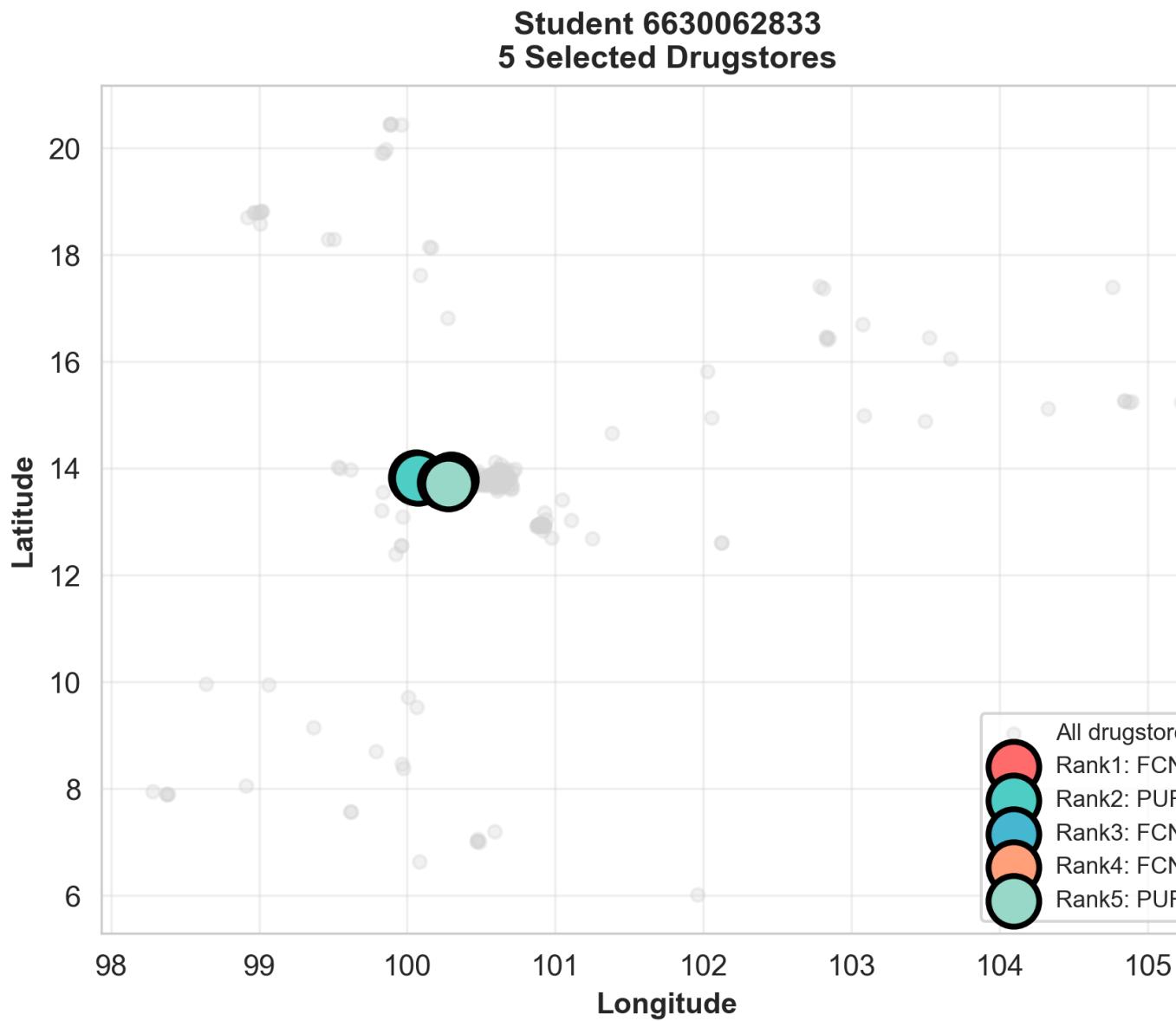
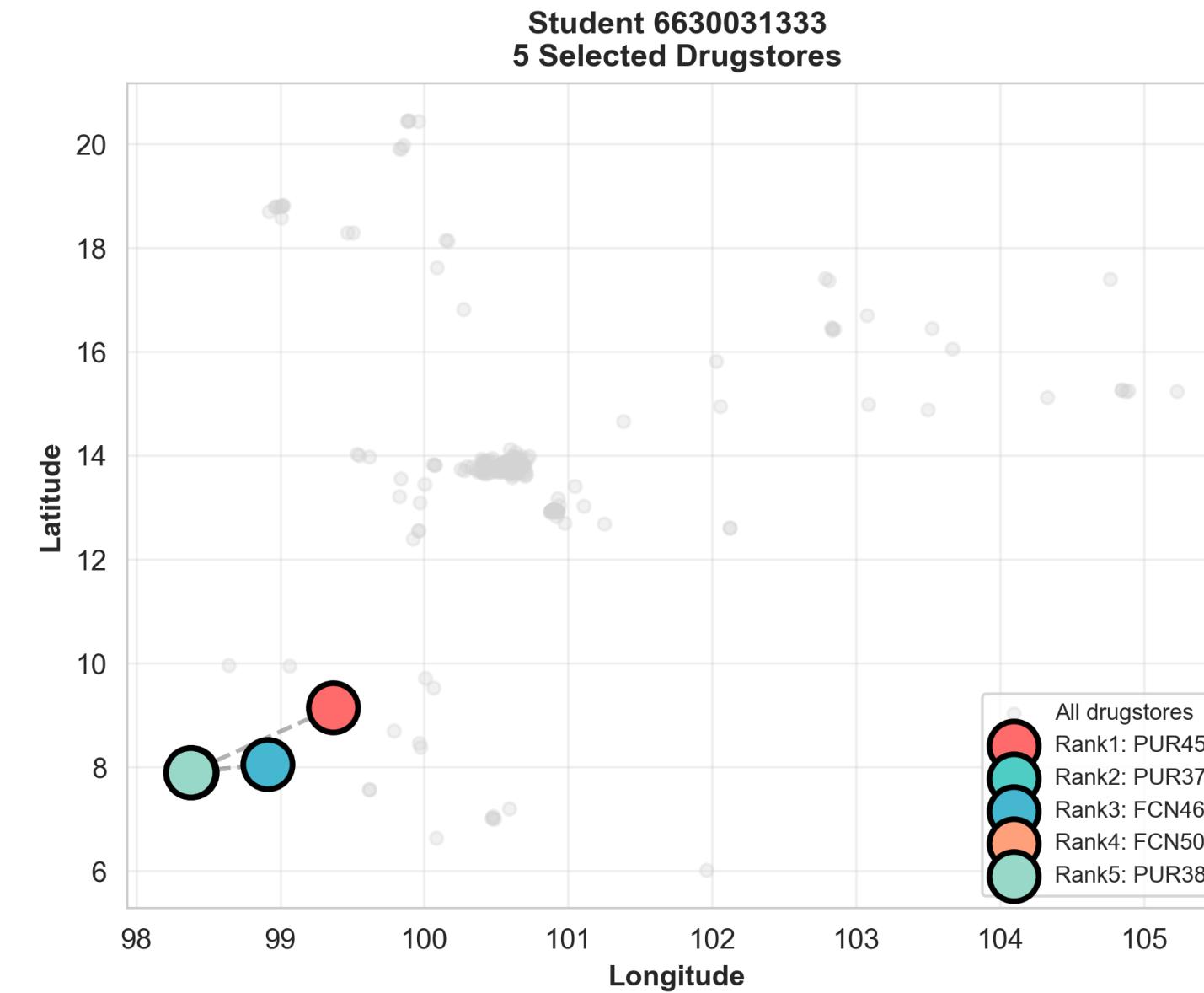
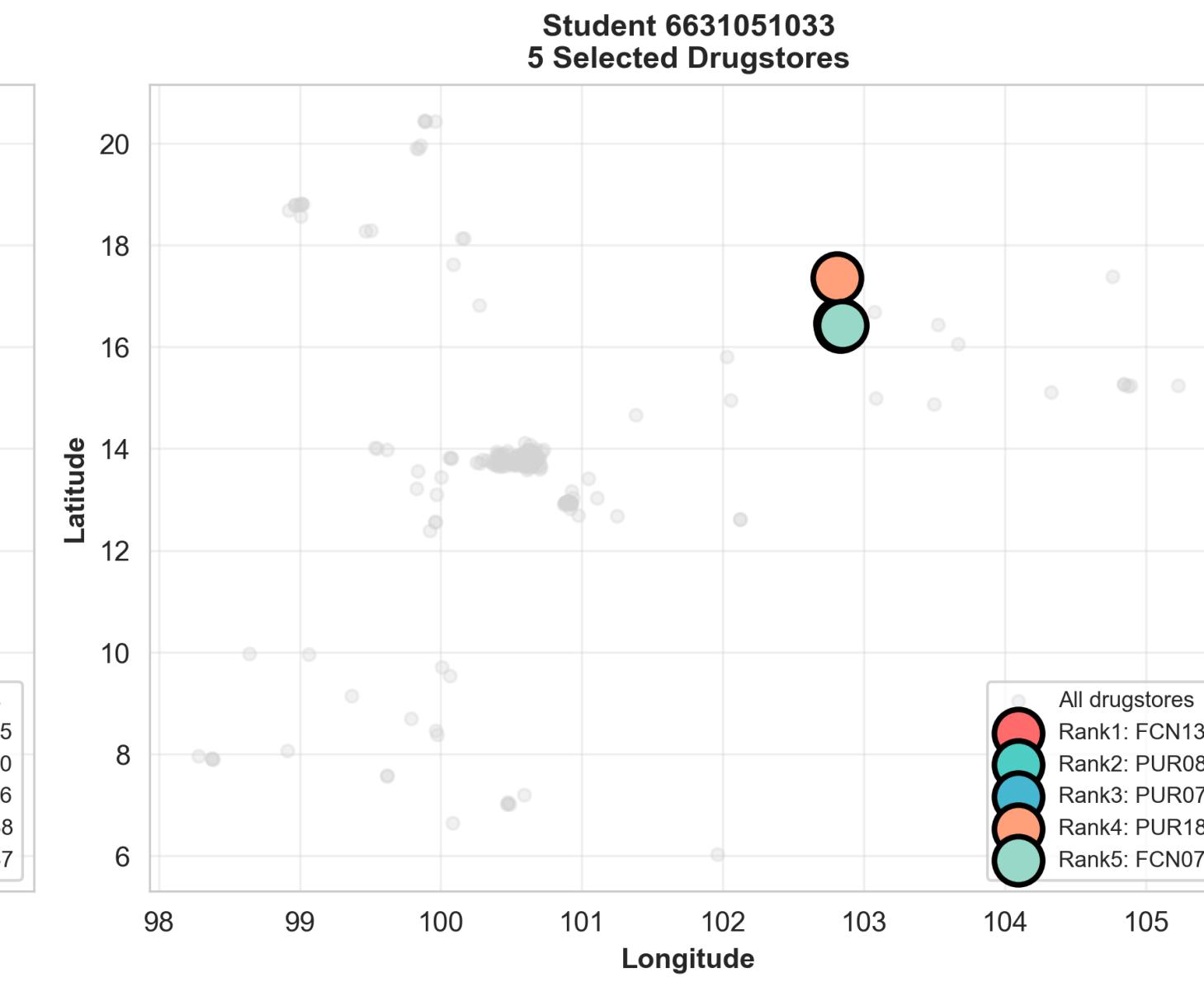
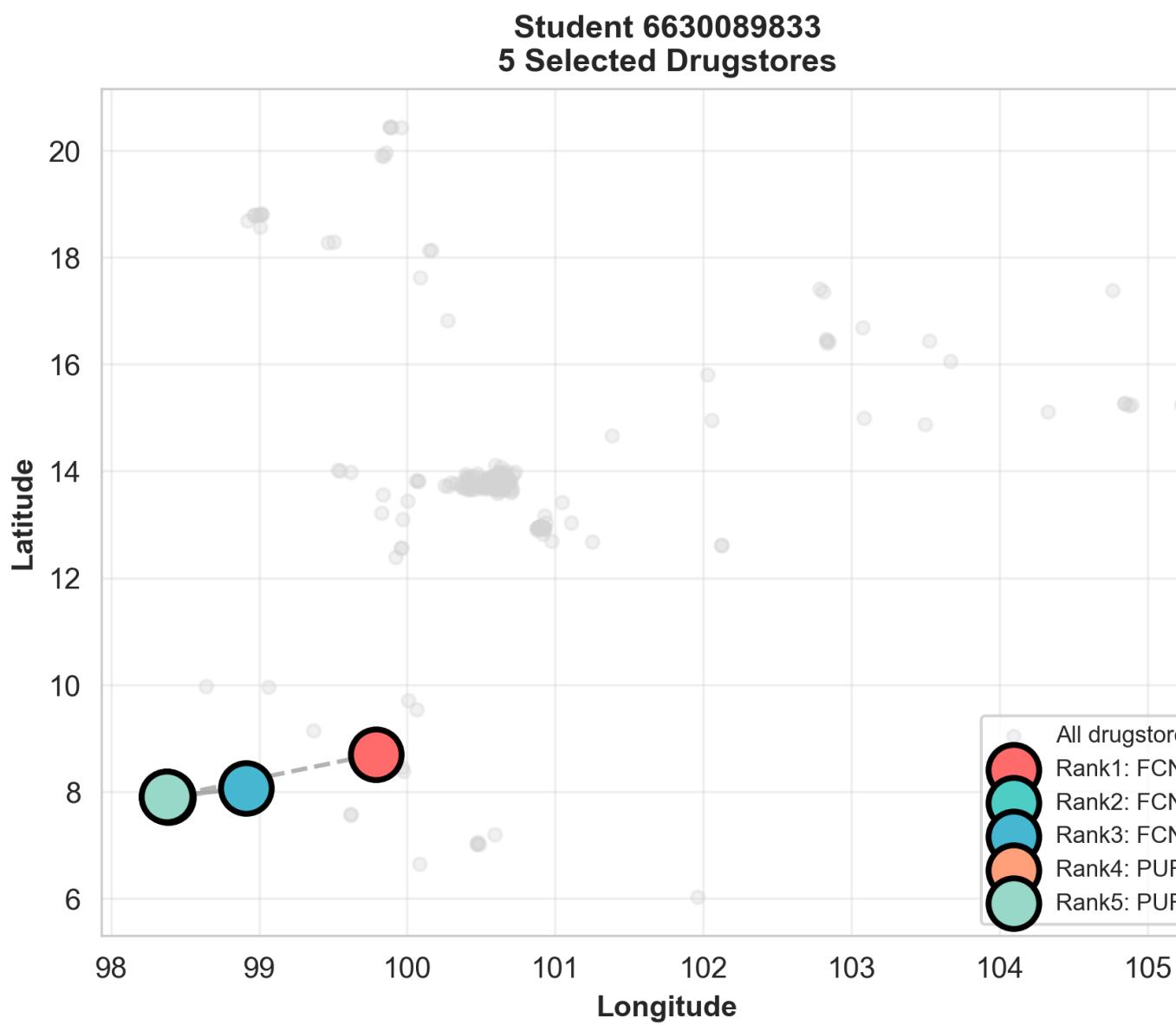
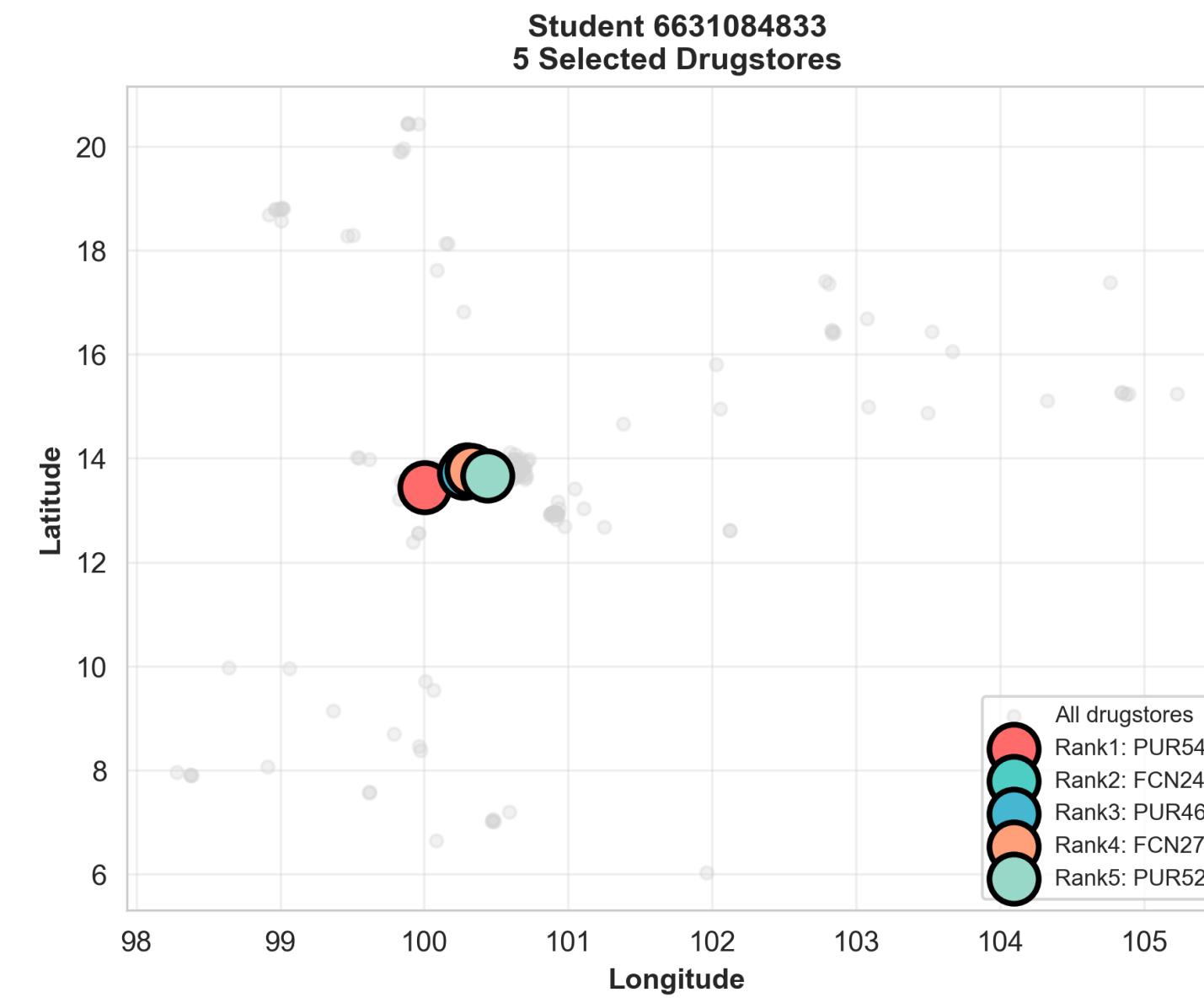
---

Selection Frequency:

- Mean per drugstore: 3.09
- Median: 3.00
- Used drugstores: 217/245
- Usage rate: 88.6%

The latitude and longitude of the sites are retrieved from Google Maps using the geopy Python library

# Sample Data / Analysis



The latitude and longitude of the sites are retrieved from Google Maps using the geopy Python library

# Sample Data / Results

Students: 134

Selected: 133 (99.3%)

Assignments by rank

**rank count**

1 113

2 14

3 5

4 0

5 1

Assignments by shift

**shift assigned students fill\_rate**

1 69 69 1.000000

2 64 65 0.984615

# Sample Forms

## Internship Placement Selection System

Student Preference Form for Internship Placement Selection System

\* Indicates required question

### Student Info

#### **Student Name \***

The student's name must be provided in English.  
(e.g. Bodin Tuesuan)

Your answer

#### **Student ID \***

e.g. 663XXXXX33

Your answer

#### **Sex \***

Choose ▾

#### **Shift \***

Choose ▾

### Ranked Internship Choices

#### **Rank 1 \***

Choose ▾

#### **Rank 2 \***

Choose ▾

#### **Rank 3 \***

Choose ▾

#### **Rank 4 \***

Choose ▾

#### **Rank 5 \***

Choose ▾