miffy-c146-s25-v03

April 4, 2025

[2]:
```
import pandas as pd
import numpy as np
```

1 Miffy: Roll the dice!

1.0.1 Data Science for Biology

Notebook developed by: *Amber Jiang* **Supervised by:** *Max Staller*

1.0.2 Learning Outcomes

In this notebook, you will learn about and practice: * Constructing over dataframes, iterating over, and manipulating them * Using dataframes to create matplotlib visualizations * Compare different types of visualizations and their strengths / weaknesses

This lab is graded manually, and has no skeleton code. You may choose to solve the problems as you wish, without importing any additional packages or using ChatGPT (and other methods prohibited in the syllabus).

2 Question 1.1

Do not start this lab until Professor Staller makes the announcement that the rolling period as ended!

As a refresher, please fill out the spreadsheet linked in Canvas with the values of each roll. The `A` column in the Google Sheet is the roll count.

Once everyone is done rolling, download the sheet as a CSV file and import it as a dataframe assigned to the variable `miffy`. Generate a output of the dataframe to understand what it looks like.

[3]:
```
miffy = pd.read_csv('miffy.csv', index_col=0)
miffy
```

[3]:
```
   Lauren Bae  Brandon Yu  Eesha Bhanot  Kaashvi Singal  Li An Annie Lin  \
1         1.0         2.0           6.0             1.0              1.0
2         3.0         4.0           6.0             2.0              4.0
3         4.0         5.0           4.0             4.0              6.0
4         5.0         1.0           4.0             2.0              4.0
```

5	3.0	6.0	3.0	3.0	3.0
6	4.0	5.0	3.0	3.0	3.0
7	1.0	6.0	5.0	4.0	1.0
8	3.0	6.0	1.0	4.0	4.0
9	5.0	6.0	5.0	6.0	1.0
10	2.0	4.0	1.0	4.0	1.0
11	4.0	6.0	2.0	4.0	5.0
12	3.0	6.0	NaN	2.0	5.0
13	6.0	4.0	NaN	4.0	2.0
14	NaN	5.0	NaN	6.0	NaN
15	NaN	5.0	NaN	4.0	NaN
16	NaN	2.0	NaN	5.0	NaN
17	NaN	2.0	NaN	NaN	NaN
18	NaN	5.0	NaN	NaN	NaN
19	NaN	3.0	NaN	NaN	NaN
20	NaN	NaN	NaN	NaN	NaN
21	NaN	NaN	NaN	NaN	NaN
22	NaN	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	NaN	NaN
24	NaN	NaN	NaN	NaN	NaN
25	NaN	NaN	NaN	NaN	NaN
26	NaN	NaN	NaN	NaN	NaN
27	NaN	NaN	NaN	NaN	NaN
28	NaN	NaN	NaN	NaN	NaN
29	NaN	NaN	NaN	NaN	NaN
30	NaN	NaN	NaN	NaN	NaN
31	NaN	NaN	NaN	NaN	NaN
32	NaN	NaN	NaN	NaN	NaN
33	NaN	NaN	NaN	NaN	NaN
34	NaN	NaN	NaN	NaN	NaN
35	NaN	NaN	NaN	NaN	NaN
36	NaN	NaN	NaN	NaN	NaN
37	NaN	NaN	NaN	NaN	NaN
38	NaN	NaN	NaN	NaN	NaN
39	NaN	NaN	NaN	NaN	NaN
40	NaN	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	NaN	NaN
42	NaN	NaN	NaN	NaN	NaN
43	NaN	NaN	NaN	NaN	NaN
44	NaN	NaN	NaN	NaN	NaN
45	NaN	NaN	NaN	NaN	NaN

	Janis Prak	Travis Gee	Heather Bastiansen	Brynne Currier	\
1	2.0	5.0	6.0	1.0	
2	1.0	2.0	3.0	5.0	
3	3.0	1.0	4.0	2.0	
4	1.0	1.0	3.0	6.0	

```
5          5.0          2.0                  4.0             1.0
6          1.0          2.0                  5.0             6.0
7          1.0          4.0                  3.0             6.0
8          5.0          6.0                  3.0             4.0
9          2.0          6.0                  3.0             5.0
10         4.0          2.0                  1.0             5.0
11         2.0          2.0                  4.0             5.0
12         1.0          2.0                  2.0             6.0
13         4.0          6.0                  NaN             4.0
14         4.0          1.0                  NaN             2.0
15         3.0          6.0                  NaN             6.0
16         5.0          6.0                  NaN             4.0
17         1.0          5.0                  NaN             5.0
18         2.0          5.0                  NaN             6.0
19         6.0          5.0                  NaN             4.0
20         NaN          6.0                  NaN             6.0
21         NaN          1.0                  NaN             1.0
22         NaN          4.0                  NaN             6.0
23         NaN          2.0                  NaN             5.0
24         NaN          3.0                  NaN             3.0
25         NaN          NaN                  NaN             NaN
26         NaN          NaN                  NaN             NaN
27         NaN          NaN                  NaN             NaN
28         NaN          NaN                  NaN             NaN
29         NaN          NaN                  NaN             NaN
30         NaN          NaN                  NaN             NaN
31         NaN          NaN                  NaN             NaN
32         NaN          NaN                  NaN             NaN
33         NaN          NaN                  NaN             NaN
34         NaN          NaN                  NaN             NaN
35         NaN          NaN                  NaN             NaN
36         NaN          NaN                  NaN             NaN
37         NaN          NaN                  NaN             NaN
38         NaN          NaN                  NaN             NaN
39         NaN          NaN                  NaN             NaN
40         NaN          NaN                  NaN             NaN
41         NaN          NaN                  NaN             NaN
42         NaN          NaN                  NaN             NaN
43         NaN          NaN                  NaN             NaN
44         NaN          NaN                  NaN             NaN
45         NaN          NaN                  NaN             NaN

    Abby Jiayi Huang  …  Grayson You  Aditya Biswal  Emma Gardner  Ryan Cho  \
1                2.0  …          5.0            6.0           2.0       3.0
2                2.0  …          2.0            3.0           1.0       2.0
3                4.0  …          6.0            1.0           4.0       1.0
4                1.0  …          6.0            1.0           1.0       2.0
```

5	4.0	…	2.0	1.0	1.0	3.0
6	5.0	…	5.0	5.0	1.0	1.0
7	4.0	…	1.0	5.0	6.0	6.0
8	5.0	…	4.0	2.0	2.0	6.0
9	3.0	…	6.0	4.0	2.0	1.0
10	1.0	…	4.0	NaN	2.0	3.0
11	3.0	…	1.0	NaN	6.0	5.0
12	5.0	…	5.0	NaN	6.0	3.0
13	2.0	…	5.0	NaN	3.0	5.0
14	4.0	…	1.0	NaN	2.0	6.0
15	3.0	…	4.0	NaN	1.0	5.0
16	2.0	…	4.0	NaN	1.0	3.0
17	4.0	…	1.0	NaN	6.0	1.0
18	3.0	…	2.0	NaN	2.0	3.0
19	2.0	…	3.0	NaN	3.0	6.0
20	3.0	…	4.0	NaN	3.0	5.0
21	5.0	…	1.0	NaN	1.0	6.0
22	5.0	…	2.0	NaN	3.0	1.0
23	1.0	…	2.0	NaN	4.0	2.0
24	6.0	…	5.0	NaN	4.0	1.0
25	NaN	…	1.0	NaN	6.0	6.0
26	NaN	…	1.0	NaN	2.0	1.0
27	NaN	…	1.0	NaN	6.0	1.0
28	NaN	…	1.0	NaN	1.0	6.0
29	NaN	…	3.0	NaN	2.0	1.0
30	NaN	…	6.0	NaN	2.0	4.0
31	NaN	…	NaN	NaN	5.0	NaN
32	NaN	…	NaN	NaN	NaN	NaN
33	NaN	…	NaN	NaN	NaN	NaN
34	NaN	…	NaN	NaN	NaN	NaN
35	NaN	…	NaN	NaN	NaN	NaN
36	NaN	…	NaN	NaN	NaN	NaN
37	NaN	…	NaN	NaN	NaN	NaN
38	NaN	…	NaN	NaN	NaN	NaN
39	NaN	…	NaN	NaN	NaN	NaN
40	NaN	…	NaN	NaN	NaN	NaN
41	NaN	…	NaN	NaN	NaN	NaN
42	NaN	…	NaN	NaN	NaN	NaN
43	NaN	…	NaN	NaN	NaN	NaN
44	NaN	…	NaN	NaN	NaN	NaN
45	NaN	…	NaN	NaN	NaN	NaN

	Qamil	Lynn Li	David M.	Eliana Romero	Christine Kim	\
1	3.0	5.0	4.0	5.0	3.0	
2	3.0	2.0	5.0	5.0	5.0	
3	6.0	4.0	3.0	3.0	3.0	
4	2.0	4.0	3.0	5.0	4.0	

5	5.0	3.0	6.0	2.0	1.0
6	3.0	6.0	4.0	1.0	2.0
7	5.0	5.0	5.0	6.0	3.0
8	4.0	3.0	1.0	1.0	2.0
9	2.0	4.0	5.0	1.0	1.0
10	6.0	1.0	3.0	6.0	3.0
11	1.0	NaN	6.0	1.0	2.0
12	NaN	NaN	3.0	3.0	6.0
13	NaN	NaN	5.0	5.0	NaN
14	NaN	NaN	5.0	1.0	NaN
15	NaN	NaN	5.0	4.0	NaN
16	NaN	NaN	5.0	NaN	NaN
17	NaN	NaN	5.0	NaN	NaN
18	NaN	NaN	1.0	NaN	NaN
19	NaN	NaN	4.0	NaN	NaN
20	NaN	NaN	2.0	NaN	NaN
21	NaN	NaN	1.0	NaN	NaN
22	NaN	NaN	5.0	NaN	NaN
23	NaN	NaN	3.0	NaN	NaN
24	NaN	NaN	1.0	NaN	NaN
25	NaN	NaN	1.0	NaN	NaN
26	NaN	NaN	5.0	NaN	NaN
27	NaN	NaN	2.0	NaN	NaN
28	NaN	NaN	3.0	NaN	NaN
29	NaN	NaN	5.0	NaN	NaN
30	NaN	NaN	5.0	NaN	NaN
31	NaN	NaN	3.0	NaN	NaN
32	NaN	NaN	NaN	NaN	NaN
33	NaN	NaN	NaN	NaN	NaN
34	NaN	NaN	NaN	NaN	NaN
35	NaN	NaN	NaN	NaN	NaN
36	NaN	NaN	NaN	NaN	NaN
37	NaN	NaN	NaN	NaN	NaN
38	NaN	NaN	NaN	NaN	NaN
39	NaN	NaN	NaN	NaN	NaN
40	NaN	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	NaN	NaN
42	NaN	NaN	NaN	NaN	NaN
43	NaN	NaN	NaN	NaN	NaN
44	NaN	NaN	NaN	NaN	NaN
45	NaN	NaN	NaN	NaN	NaN

	Srivishal Sudharsan
1	3.0
2	5.0
3	2.0
4	4.0

```
5                    2.0
6                    5.0
7                    6.0
8                    4.0
9                    2.0
10                   2.0
11                   6.0
12                   4.0
13                   6.0
14                   4.0
15                   2.0
16                   3.0
17                   5.0
18                   3.0
19                   5.0
20                   1.0
21                   NaN
22                   NaN
23                   NaN
24                   NaN
25                   NaN
26                   NaN
27                   NaN
28                   NaN
29                   NaN
30                   NaN
31                   NaN
32                   NaN
33                   NaN
34                   NaN
35                   NaN
36                   NaN
37                   NaN
38                   NaN
39                   NaN
40                   NaN
41                   NaN
42                   NaN
43                   NaN
44                   NaN
45                   NaN

[45 rows x 34 columns]
```

3 Question 1.2

A more intuitive way of displaying the data would be having student names be indices and the columns being the roll count. You may accomplish this through a number of ways.

After transposing the `miffy` dataframe, assign the new dataframe to the `miffy2` variable. Again, your indices should be student names and the columns should be the roll or toss number.

Take a look at the new dataframe now.

[4]:
```
miffy2 = miffy.transpose()
miffy2.head()
```

[4]:
```
                  1    2    3    4    5    6    7    8    9    10  …  36  \
Lauren Bae       1.0  3.0  4.0  5.0  3.0  4.0  1.0  3.0  5.0  2.0  … NaN
Brandon Yu       2.0  4.0  5.0  1.0  6.0  5.0  6.0  6.0  6.0  4.0  … NaN
Eesha Bhanot     6.0  6.0  4.0  4.0  3.0  3.0  5.0  1.0  5.0  1.0  … NaN
Kaashvi Singal   1.0  2.0  4.0  2.0  3.0  3.0  4.0  4.0  6.0  4.0  … NaN
Li An Annie Lin  1.0  4.0  6.0  4.0  3.0  3.0  1.0  4.0  1.0  1.0  … NaN

                  37  38  39  40  41  42  43  44  45
Lauren Bae       NaN NaN NaN NaN NaN NaN NaN NaN NaN
Brandon Yu       NaN NaN NaN NaN NaN NaN NaN NaN NaN
Eesha Bhanot     NaN NaN NaN NaN NaN NaN NaN NaN NaN
Kaashvi Singal   NaN NaN NaN NaN NaN NaN NaN NaN NaN
Li An Annie Lin NaN NaN NaN NaN NaN NaN NaN NaN NaN

[5 rows x 45 columns]
```

You may notice many `NaN` values in the dataframe. Do not replace those values or delete those cells as you work. Instead, you should account for their existence in your code, in the sense that your calculations or dataframe manipulations should not consider `NaN` as a relevant value.

Preserving `NaN` values in the real world can be informative and important, even when they're not used as part of calculations. Part of the challenge in this lab is to handle them appropriately.

4 Question 2.1

Use the `miffy2` dataframe to generate a new dataframe, `unique_rolls`, that also has students as the indices, but the column names are the number of unique rolls. The data within each cell represents how many rolls it took each student to achieve a specific number of unique rolls.

Hint: everyone should have taken exactly 1 roll to get 1 unique roll (first column). How many unique rolls are possible? Does the largest value in the cells make sense?

[5]:
```
# turns a row into a list of 6 values of when a new unique value was found
def count_unique(student):
    unique = set()
    rolls = []
    for i in range(len(student)):
```

```python
        value = student[i]
        if not pd.notna(value):
            continue
        if value not in unique:
            unique.add(value)
            rolls.append(i+1)
    return rolls

# creates a matrix, applying the function to each row
new_data = []
for index, student in miffy2.iterrows():
    new_data.append(count_unique(student.tolist()))

# turning matrix into dataframe
unique_rolls = pd.DataFrame(new_data, index = miffy2.index, columns=[1,2,3,4,5,6])
unique_rolls.head()
```

[5]:

	1	2	3	4	5	6
Lauren Bae	1	2	3	4	10	13
Brandon Yu	1	2	3	4	5	19
Eesha Bhanot	1	3	5	7	8	11
Kaashvi Singal	1	2	3	5	9	16
Li An Annie Lin	1	2	3	5	11	13

5 Question 2.2

Use the `miffy2` dataframe to generate a new dataframe, `unique_count`, that also has students as the indicies, and also has the column names as the roll count, but has the data within each cell represent how many unique rolls have been made.

Hint: no cell (not including column names) should have a value above 6.

[6]:
```python
# can also start by initializing the new dataframe
unique_counts = pd.DataFrame(index = miffy2.index, columns = miffy2.columns)

for index, student in miffy2.iterrows():
    unique = set()
    unique_counts_values = []

    for roll in student:
        if pd.notna(roll):
            unique.add(roll)
        unique_counts_values.append(len(unique))

    unique_counts.loc[index] = unique_counts_values
```

```
unique_counts.head()
```

[6]:
```
                1  2  3  4  5  6  7  8  9  10  … 36 37 38 39 40 41 42 43 44  \
Lauren Bae       1  2  3  4  4  4  4  4  4  5  …  6  6  6  6  6  6  6  6  6
Brandon Yu       1  2  3  4  5  5  5  5  5  5  …  6  6  6  6  6  6  6  6  6
Eesha Bhanot     1  1  2  2  3  3  4  5  5  5  …  6  6  6  6  6  6  6  6  6
Kaashvi Singal   1  2  3  3  4  4  4  4  5  5  …  6  6  6  6  6  6  6  6  6
Li An Annie Lin  1  2  3  3  4  4  4  4  4  4  …  6  6  6  6  6  6  6  6  6

                45
Lauren Bae       6
Brandon Yu       6
Eesha Bhanot     6
Kaashvi Singal   6
Li An Annie Lin  6

[5 rows x 45 columns]
```

6 Question 3.1

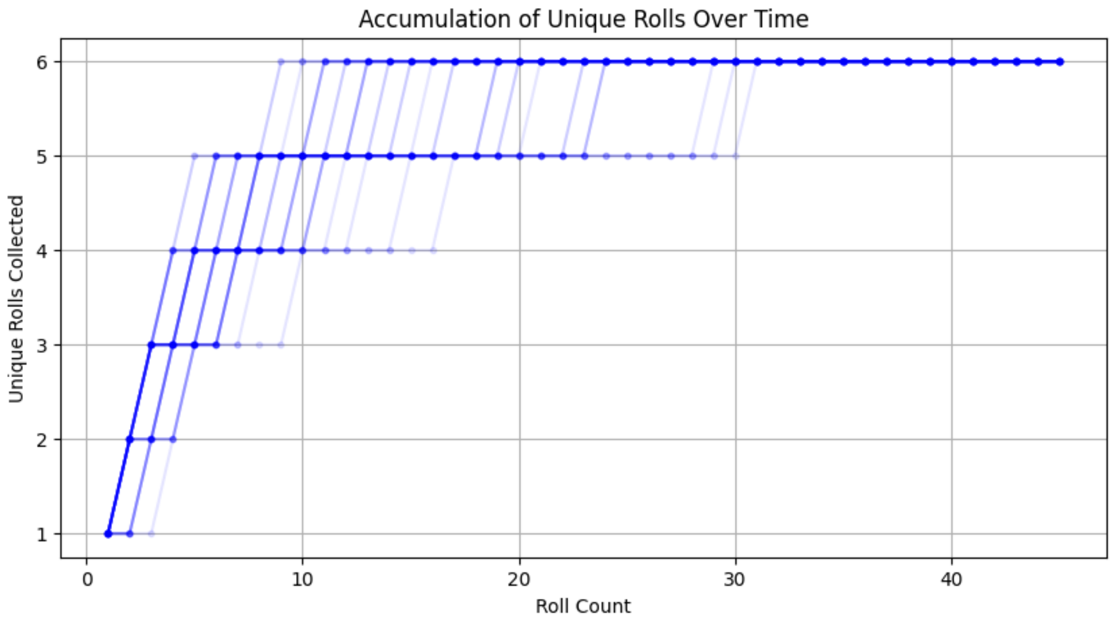
It's time to begin visualizing! For your first graph, create a line graph using `unique_counts`. Each student should have their own line.

Your x-axis should be the roll count, and your y-axis should be the unique number of rolls accomplished by each student. Edit the density or transparency of each line so that the overall density in the graph represents how the prevalence of unique rolls at each roll count. Remember that as lines overlap, their density is summative.

Please remember to label your graph as appropriate! Use your best judgement when it comes to color, formatting, and legends.

[7]:
```python
import matplotlib.pyplot as plt
```

[8]:
```python
# Plot
plt.figure(figsize=(10, 5))
for idx, row in unique_counts.iterrows():
    plt.plot(row.index, row.values, marker=".", alpha = 0.1, linestyle="-",␣
 ↪label=idx, color = "blue")

plt.xlabel("Roll Count")
plt.ylabel("Unique Rolls Collected")
plt.title("Accumulation of Unique Rolls Over Time")
plt.grid(True)
plt.show()
```



6.0.1 Free response question: around when do the majority of students manage to roll all 6 unique values? It's okay to provide a rough estimation.

Based on the above graph it looks like this happens after about 10 rolls of the dice

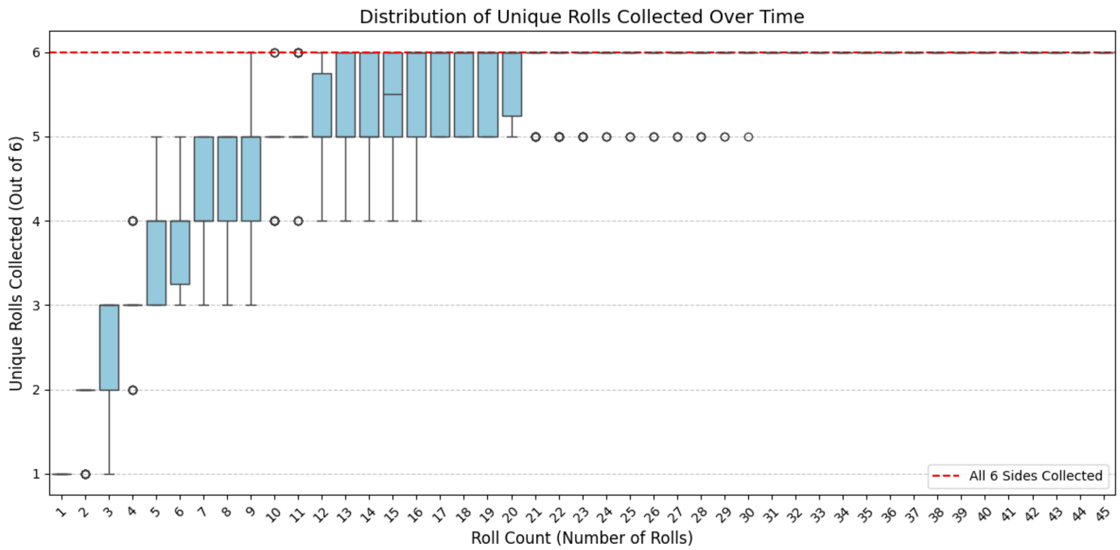
7 Question 3.2

There is a lot overlapping data in this last plot, which can make it tricky to appreciate what is going on. Experiment with another way of plotting the data to better capture the overlapping nature of the data, that is not a histogram (since that is question 3.3).

This is a freeform question. You can consult the galleries from matplotLib or seaborn for inspiration.

[25]:
```
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.boxplot(data=unique_counts, color="skyblue", showfliers=True, linewidth=1)



plt.axhline(y=6, color='red', linestyle='--', linewidth=1.5, label='All 6 Sides␣
↪Collected')

# plot seettings hereee
plt.xlabel("Roll Count (Number of Rolls)", fontsize=12)
plt.ylabel("Unique Rolls Collected (Out of 6)", fontsize=12)
plt.title("Distribution of Unique Rolls Collected Over Time", fontsize=14)
plt.xticks(rotation=45, fontsize=10)
```

```
plt.yticks(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend()
plt.tight_layout()
plt.show()
```



7.0.1 Free response question: what new insight(s) does your graph capture that the previous line graph failed to do?

Well the nice thing about boxplots is that we now have an idea of the spread, median, quartiles and outliers for the number of unique sides collected for each roll count. Relative to the line graph which made it difficult to understand the central tendency of our data.

We also see that our variability seems to decrease and eventually converge towards collecting all 6 sides too!

8 Question 3.3

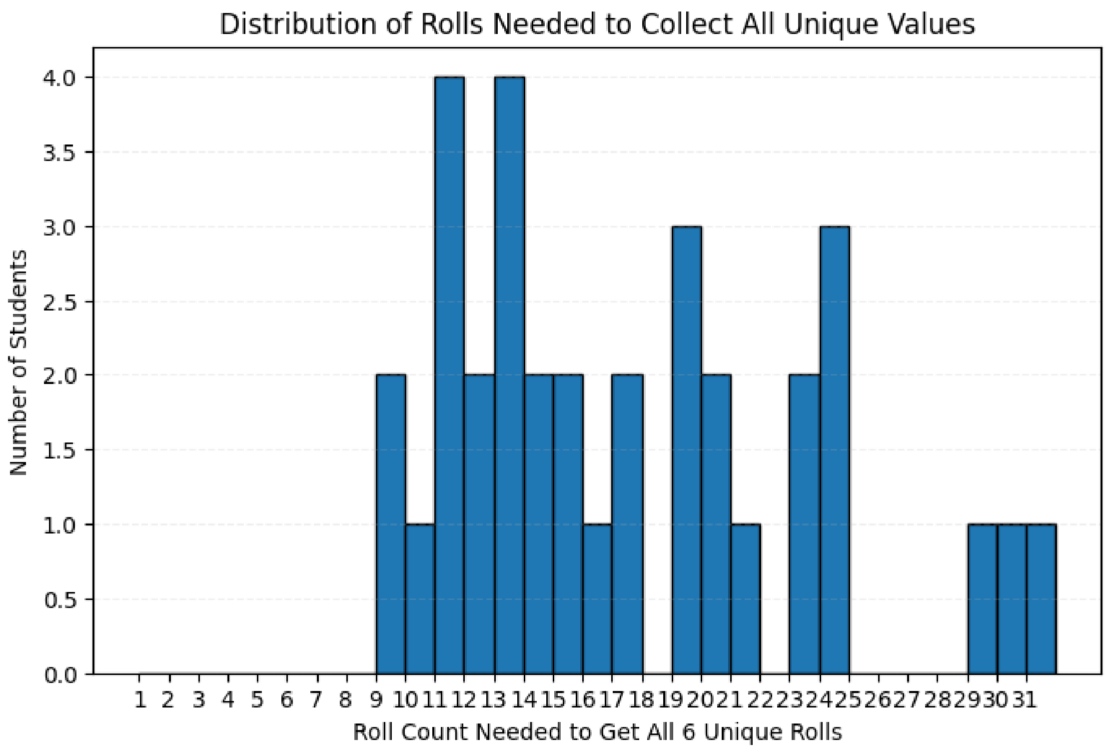
For your second graph, create a histogram also using `unique_counts`.

Your x-axis should be the roll count needed to get all 6 unique values, and your y-axis should now be the number of students.

Please remember to label your graph as appropriate! Use your best judgement when it comes to color, formatting, and legends.

[20]:
```
# Find the roll number where each student first reaches 6 unique rolls
rolls_needed = unique_counts.eq(6).idxmax(axis=1).astype(int)

# Plot histogram
```

```
plt.figure(figsize=(8, 5))
plt.hist(rolls_needed, bins=range(1, rolls_needed.max() + 2),␣
 ↪edgecolor='black', alpha=1)
plt.xlabel("Roll Count Needed to Get All 6 Unique Rolls")
plt.ylabel("Number of Students")
plt.title("Distribution of Rolls Needed to Collect All Unique Values")
plt.xticks(range(1, rolls_needed.max() + 1))
plt.grid(axis="y", linestyle="--", alpha=0.2)
plt.show()
```



8.0.1 Free response question: which roll count(s) did the most students need to get all 6 unique values? Which was the least?

It looks like the most common roll counts are tied between 11, 13 and 14 rolls with 4 students each reaching all 6 unique rolls at these counts. On the flipside, the least common roll counts were 10, 16, 21, 29, 30, and 31 with only one student each needing these roll counts!

Congrats! You're done with the Miffy lab. We hope you've had fun! There is no autograder or zip file needed with this assignment. Just upload both Jupyter Notebook file and PDF to GradeScope like in the earlier weeks.

---

9 Extra Credit

Take a group of classmates and repeat this exercise with dice that don't have 6 sides ("d6s"). Ask your Dungeons & Dragons player friends, who may have some d4s to d20s lying around that you can use. (If you want a particularly difficult and draining challenge, you may try this with a deck of cards and trying to draw each unique value.)

Roll the dice, record your rolls, and run the calculations again. Try some different ways to visualize and analyze the data. Feel free to use seaborn for the extra credit portion.

Note any interesting observations, such as comparing and contrasting the descriptive statistics between data from different dice.

Please submit a separate notebook to the Miffy extra credit assignment in Gradescope. In your assignment, add the names of the team members in your group. Please also submit on Gradescope as a group project, adding in your team members.