

# Midterm Exam – DSC 10, Winter 2022

## Q1 Skyscrapers

4 Points

Welcome to the Midterm Exam for DSC 10 this quarter! Throughout this exam, we will work with a dataset consisting of various skyscrapers in the US, which we've loaded into a DataFrame called `sky`. The first few rows of `sky` are shown below (though the full DataFrame has more rows):

	material	city	floors	height	year
name					
<b>Bayard-Condict Building</b>	steel	New York City	13	49.380001	1899
<b>The Yacht Club at Portofino</b>	concrete	Miami Beach	33	103.900002	1999
<b>City Investing Building</b>	steel	New York City	33	148.289993	1908
<b>Solitair Brickell</b>	concrete	Miami	48	169.199997	2017
<b>Esquire Plaza</b>	steel	Sacramento	22	108.199997	1999

Each row of `sky` corresponds to a single skyscraper. For each skyscraper, we have:

- its name, which is stored in the index of `sky` (string)
- the `'material'` it is made up of (string)
- the `'city'` in the US where it is located (string)
- the number of `'floors'` (levels) it contains (int)
- its `'height'` in meters (float), and
- the `'year'` in which it was opened (int)

Note that the height of a floor may be different in each building.

**Below, identify the data type of the result of each of the following expressions, or select "error" if you believe the expression results in an error.**

### Q1.1

0.5 Points

```
sky.sort_values('height')
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.2

0.5 Points

```
sky.sort_values('height').get('material').loc[0]
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.3

0.5 Points

```
sky.sort_values('height').get('material').iloc[0]
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.4

0.5 Points

```
sky.get('city').apply(len)
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.5

0.5 Points

```
sky.get('city').apply(max)
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.6

0.5 Points

```
sky.get('floors').max()
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.7

0.5 Points

```
sky.groupby('material').max()
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

### Q1.8

0.5 Points

```
sky.index[0]
```

- ☐ int or float
- ☐ Boolean
- ☐ string
- ☐ array
- ☐ Series
- ☐ DataFrame
- ☐ error

## Q2 San Diego 🌞

3 Points

**Note that each part of Question 2 depends on previous parts of Question 2.**

For your convenience, we show the first few rows of `sky` once again.

name	material	city	floors	height	year
Bayard-Condict Building	steel	New York City	13	49.380001	1899
The Yacht Club at Portofino	concrete	Miami Beach	33	103.900002	1999
City Investing Building	steel	New York City	33	148.289993	1908
Solitair Brickell	concrete	Miami	48	169.199997	2017
Esquire Plaza	steel	Sacramento	22	108.199997	1999

In this question, we'll write code to learn more about the skyscrapers in the beautiful city of San Diego. (Unrelated fun fact – since the San Diego Airport is so close to downtown, buildings in downtown San Diego legally cannot be taller than 152 meters.)

### Q2.1

1 Point

Below, fill in the blanks to create a DataFrame, named `san_tall`, consisting of just the skyscrapers in San Diego that are over 100 meters tall.

```
condition = __ (a) __  
san_tall = sky[(sky.get('city') == 'San Diego') & condition]
```

What goes in blank (a)?

## Q2.2

1 Point

Suppose `san_tall` from the previous part (2.1) was created correctly. Fill in the blanks so that `height_many_floors` evaluates to the **height (in meters)** of the skyscraper with the **most floors**, amongst all skyscrapers in San Diego that are over 100 meters tall.

```
height_many_floors = san_tall.__(a)__.iloc[0]
```

What goes in blank (a)?

## Q2.3

1 Point

`height_many_floors`, the value you computed in the previous part (2.2) was a number.

**True or False:** Assuming that the DataFrame `san_tall` contains all skyscrapers in San Diego, `height_many_floors` is the height (in meters) of the **tallest** skyscraper in San Diego.

- ☐ True
- ☐ False

## Q3 Concrete

6 Points

Note that each part of Question 3 depends on previous parts of Question 3.

For your convenience, we show the first few rows of `sky` once again.

name	material	city	floors	height	year
Bayard-Condict Building	steel	New York City	13	49.380001	1899
The Yacht Club at Portofino	concrete	Miami Beach	33	103.900002	1999
City Investing Building	steel	New York City	33	148.289993	1908
Solitair Brickell	concrete	Miami	48	169.199997	2017
Esquire Plaza	steel	Sacramento	22	108.199997	1999

In this question, we'll take a closer look at the `'material'` column of `sky`.

### Q3.1

2 Points

Below, fill in the blank to complete the implementation of the function `majority_concrete`, which takes in the name of a `city` and returns `True` if the majority of the skyscrapers in that city are made of concrete, and `False` otherwise. **We define "majority" to mean "at least 50%".**

```
def majority_concrete(city):  
    all_city = sky[sky.get('city') == city]  
    concrete_city = all_city[all_city('material') == 'concrete']  
    proportion = __ (a) __  
    return proportion >= 0.5
```

What goes in blank (a)?



### Q3.2

1 Point

Below, we create a DataFrame named `by_city`.

```
by_city = sky.groupby('city').count().reset_index()
```

Below, fill in the blanks to add a column to `by_city`, called `'is_majority'`, that contains the value `True` for each city where the majority of skyscrapers are concrete, and `False` for all other cities. You may need to use the function you defined in 3.1.

```
by_city = by_city.assign(is_majority = __ (a) __)
```

What goes in blank (a)?

### Q3.3

1 Point

`by_city` now has a column named `'is_majority'` as described in the previous part (3.2). Now, suppose we create another DataFrame, `mystery`, below:

```
mystery = by_city.groupby('is_majority').count()
```

What is the largest possible value that `mystery.shape[0]` could evaluate to?

### Q3.4

2 Points

Suppose `mystery.get('city').iloc[0] == mystery.get('city').iloc[1]` evaluates to `True`.

**True or False:** In exactly half of the cities in `sky`, it is true that a majority of skyscrapers are made of concrete. (**Tip:** Walk through the manipulations performed in 3.1, 3.2, and 3.3 to get an idea of what `mystery` looks like and contains.)

☐ True

☐ False

## Q4 The Big Apple 🍏

2 Points

Suppose we have access to another DataFrame, `new_york`, that contains the latitude and longitude of every single skyscraper in New York City that is also in `sky`. The first few rows of `new_york` are shown below.

	name	latitude	longitude
0	One World Trade Center	40.713112	-74.013351
1	Central Park Tower	40.766361	-73.980949
2	World Trade Building	40.759258	-73.989471
3	111 West 57th Street	40.764801	-73.977547
4	432 Park Avenue	40.761559	-73.971863

Below, we define a new DataFrame, `sky_with_location`, that merges together both `sky` and `new_york`.

```
sky_with_location = sky.merge(new_york, left_index=True, right_on='name')
```

Given that:

- `sky` has  $s$  rows,
- `new_york` has  $n$  rows, and
- building names are spelled and formatted the exact same way in both `sky` and `new_york`, i.e. that there are no typos in either DataFrame,

select the true statement below.

- ☐ `sky_with_location` has exactly  $s$  rows.
- ☐ `sky_with_location` has exactly  $n$  rows.
- ☐ `sky_with_location` has exactly  $s - n$  rows.
- ☐ `sky_with_location` has exactly  $s + n$  rows.
- ☐ `sky_with_location` has exactly  $s \times n$  rows.

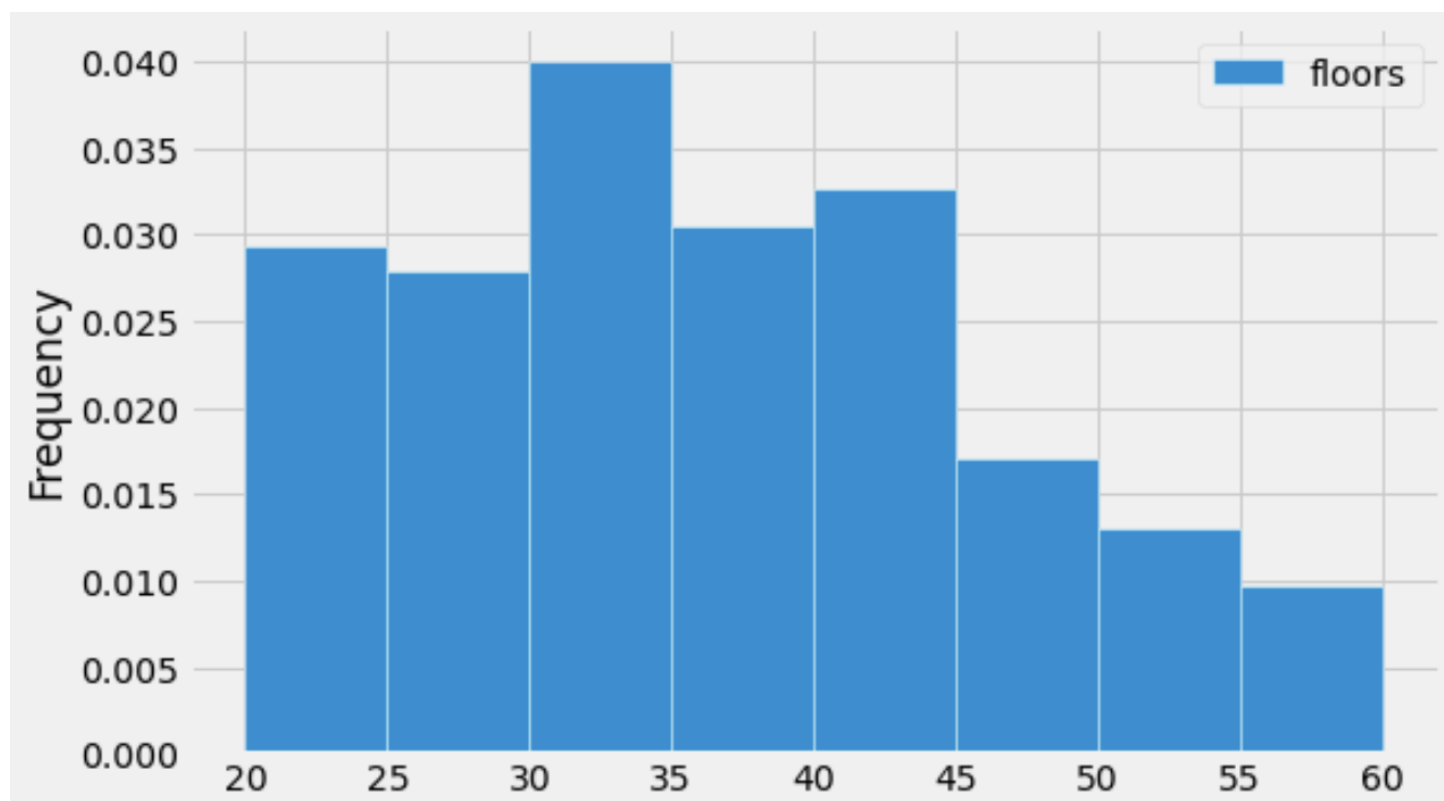
## Q5 On the Floor (feat. Pitbull) 🏆

4 Points

**Note that each part of Question 5 depends on previous parts of Question 5.**

Recall, the interval  $[a, b)$  refers to numbers greater than or equal to  $a$  and less than  $b$ , and the interval  $[a, b]$  refers to numbers greater than or equal to  $a$  and less than or equal to  $b$ .

Suppose we created a DataFrame, `medium_sky`, containing only the skyscrapers in `sky` whose number of floors are in the interval  $[20, 60]$ . Below, we've drawn a histogram of the number of floors of all skyscrapers in `medium_sky`.



### Q5.1

2 Points

Suppose that there are 160 skyscrapers whose number of floors are in the interval  $[30, 35)$ .

Given this information and the histogram above, how many skyscrapers are there in `medium_sky`?

## Q5.2

2 Points

Again, suppose that there are 160 skyscrapers whose number of floors are in the interval  $[30, 35)$ .

Now suppose that there is a typo in the `medium_sky` DataFrame, and 20 skyscrapers were accidentally listed as having 53 floors each when instead they actually only have 35 floors each. The histogram drawn above contains the incorrect version of the data.

Suppose we re-draw the above histogram using the correct data. What will be the new heights of both the  $[35, 40)$  bar and  $[50, 55)$  bar? Select the closest answer.

- ☐ The  $[35, 40)$  bar's height becomes 0.0325, and the  $[50, 55)$  bar's height becomes 0.0105.
- ☐ The  $[35, 40)$  bar's height becomes 0.035, and the  $[50, 55)$  bar's height becomes 0.008.
- ☐ The  $[35, 40)$  bar's height becomes 0.0375, and the  $[50, 55)$  bar's height becomes 0.0055.
- ☐ The  $[35, 40)$  bar's height becomes 0.04, and the  $[50, 55)$  bar's height becomes 0.003.

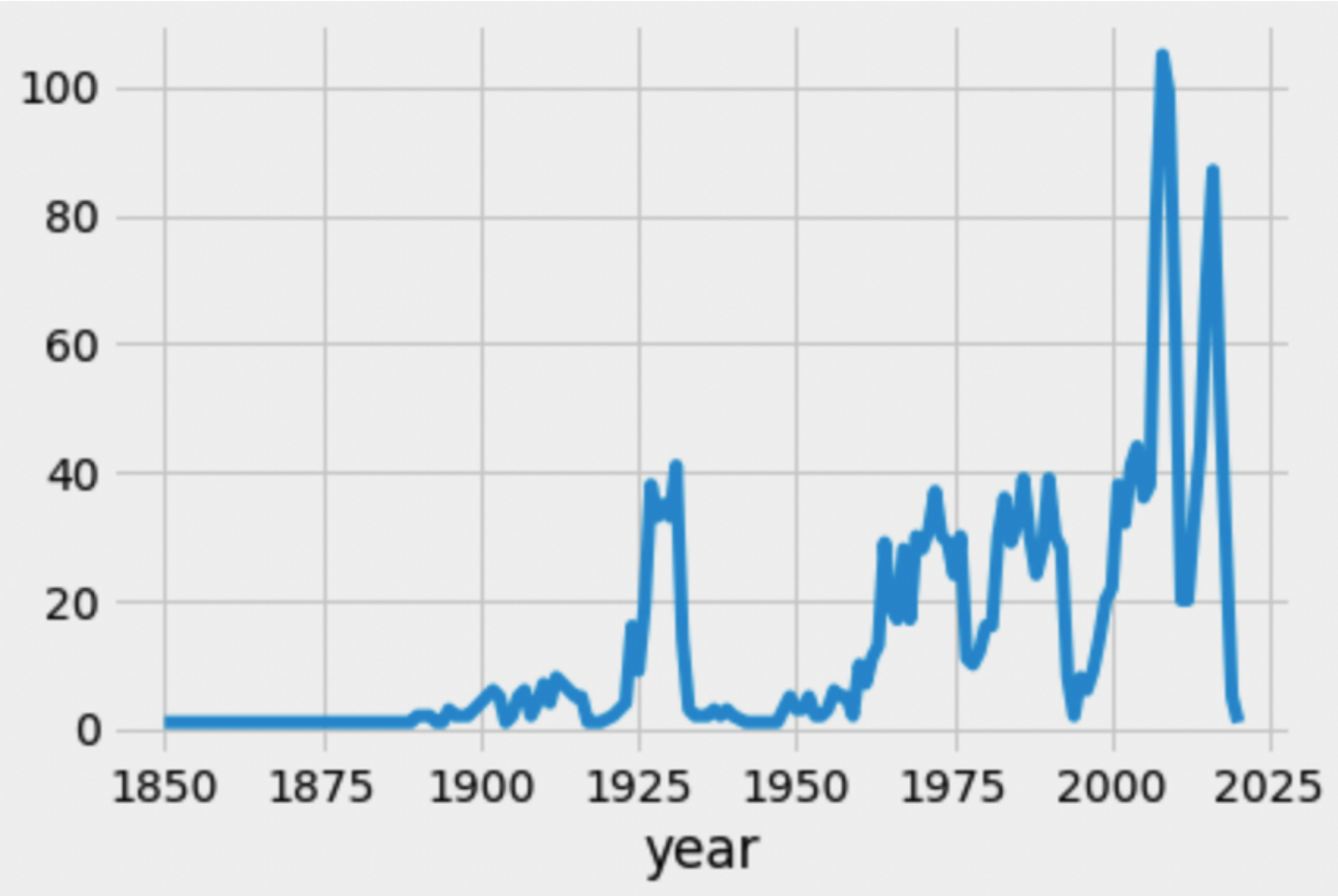
Q6 Time is Money 💰

4 Points

For your convenience, we show the first few rows of `sky` again below.

name	material	city	floors	height	year
Bayard-Condict Building	steel	New York City	13	49.380001	1899
The Yacht Club at Portofino	concrete	Miami Beach	33	103.900002	1999
City Investing Building	steel	New York City	33	148.289993	1908
Solitair Brickell	concrete	Miami	48	169.199997	2017
Esquire Plaza	steel	Sacramento	22	108.199997	1999

Now consider the following line plot, which depicts the number of skyscrapers built per year.



## Q6.1

2 Points

We created the line plot above using the following line of code:

```
sky.groupby('year').count().plot(kind='line', y='height');
```

Which of the following could we replace `'height'` with in the line of code above, such that the resulting line of code creates the same line plot? **Select all that apply.**

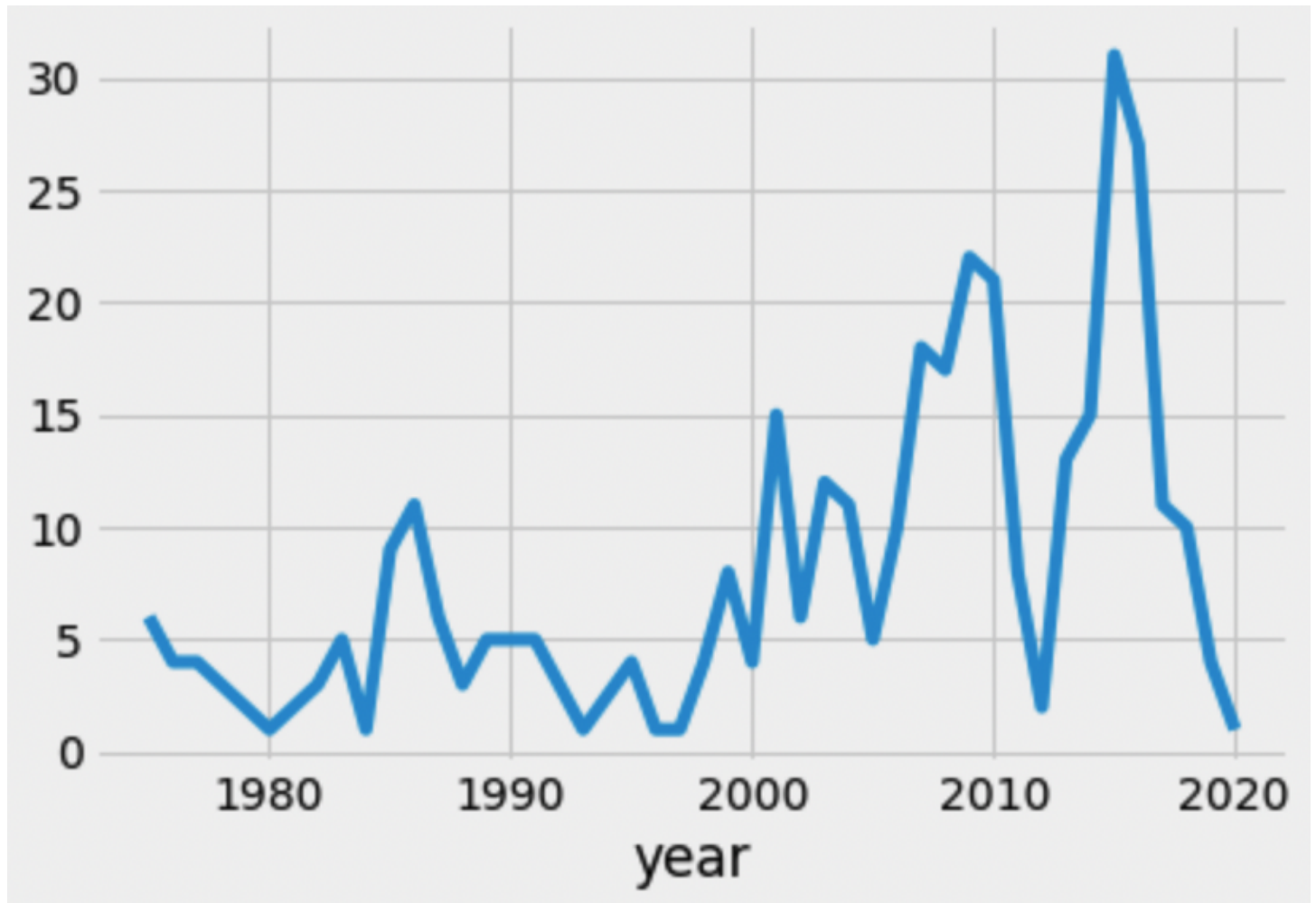
- ☐ `'name'`
- ☐ `'material'`
- ☐ `'city'`
- ☐ `'floors'`
- ☐ `'year'`
- ☐ None of the above



## Q6.2

1 Point

Now let's look at the number of skyscrapers built each year since 1975 in New York City 🗽.



Which of the following is a valid conclusion we can make using this graph alone?

- ☐ No city in the dataset had more skyscrapers built in 2015 than New York City.
- ☐ The decrease in the number of skyscrapers built in 2012 over previous years was due to the 2008 economic recession, and the reason the decrease is seen in 2012 rather than 2008 is because skyscrapers usually take 4 years to be built.
- ☐ The decrease in the number of skyscrapers built in 2012 over previous years was due to something other than the 2008 economic recession.
- ☐ The COVID-19 pandemic is the reason that so few skyscrapers were built in 2020.
- ☐ None of the above.

### Q6.3

1 Point

In which of the following scenarios would it make sense to draw a overlaid histogram?

- ☐ To visualize the number of skyscrapers of each material type, separately for New York City and Chicago.
- ☐ To visualize the distribution of the number of floors per skyscraper, separately for New York City and Chicago.
- ☐ To visualize the average height of skyscrapers built per year, separately for New York City and Chicago.
- ☐ To visualize the relationship between the number of floors and height for all skyscrapers.

## **Q7** Oh No...

3 Points

**Note that each part of Question 7 depends on previous parts of Question 7.**

Billina Records, a new record company focused on creating new TikTok audios, has its offices on the 23rd floor of a skyscraper with 75 floors (numbered 1 through 75). The owners of the building promised that 10 different random floors will be selected to be renovated.

## Q7.1

2 Points

Below, fill in the blanks to complete a simulation that will estimate the probability that Billina Records' floor will be renovated.

```
total = 0
repetitions = 10000
for i in np.arange(repetitions):
    choices = np.random.choice(__(a)__, 10, __(b)__)
    if __(c)__:
        total = total + 1
prob_renovate = total / repetitions
```

What goes in blank (a)?

- ☐ `np.arange(1, 75)`
- ☐ `np.arange(10, 75)`
- ☐ `np.arange(0, 76)`
- ☐ `np.arange(1, 76)`

What goes in blank (b)?

- ☐ `replace=True`
- ☐ `replace=False`

What goes in blank (c)?

- ☐ `choices == 23`
- ☐ `choices is 23`
- ☐ `np.count_nonzero(choices == 23) > 0`
- ☐ `np.count_nonzero(choices) == 23`
- ☐ `choices.str.contains(23)`

## Q7.2

1 Point

In the previous part of this question, your answer to blank (c) contained the number 23, and the simulated probability was stored in the variable `prob_renovate`.

Suppose, in blank (c), we change the number 23 to the number 46, and we store the new simulated probability in the variable name `other_prob`. (`prob_renovate` is unchanged from the previous part.)

With these changes, which of the following is the most accurate representation of the relationship between `other_prob` and `prob_renovate`?

- ☐ `other_prob` will be roughly half of `prob_renovate`
- ☐ `other_prob` will be roughly equal to `prob_renovate`
- ☐ `other_prob` will be roughly double `prob_renovate`

## Q8 Cat in the Hat 🐱

4 Points

While they are not skyscrapers, New Sixth College at UCSD has four relatively tall residential buildings, which we'll call Building A, Building B, Building C, and Building D. Suppose each building has 10 floors.

Sixth College administration decides to ease the General Education requirements for a few randomly selected students. Here's their strategy:

- **Wave 1:** Select, at random, one floor from each building.
- **Wave 2:** Select, at random, one of the four floors that was selected in Wave 1.

Everyone on one of the four floors selected in Wave 1 has the CAT 1 requirement waived. Everyone on the one floor selected in Wave 2 has both the CAT 1 and CAT 2 requirements waived.

### Q8.1

2 Points

Billy lives on the 8th floor of Building C. What's the probability that Billy has both the CAT 1 and CAT 2 requirements waived? Give your answer as a proportion between 0 and 1, rounded to 3 decimal places.

### Q8.2

2 Points

What's the probability that **at least one** of the top (10th) floors of all four buildings are selected in Wave 1?

Give your answer as a proportion between 0 and 1, rounded to 3 decimal places.

---