Formal Analyses (CS50) Assignment 2

Variables with LBA

Overview

This assignment is intended to be a location based-assignment that will require you to interact with the city around in you in a new way. You will identify a measurable variable in the city and then create an estimate using the Fermi estimation technique. Next, you will complete the data collection, calculate descriptive statistics on the data, and create relevant data visualizations. You will also have a chance to apply your knowledge of probability and simulation to solve a problem.

You'll notice several *Optional* problems throughout the assignment to challenge yourself. These will only be scored if they are completed correctly with thorough explanation. If you attempt an optional challenge but do not succeed, you will not be penalized with a low score.

This is an individual assignment. Everything you submit should be your own words and reflect your own understanding of the material.

Formatting & HC Guidelines

You must complete all tasks within this pre-formatted Jupyter notebook. **Please follow the formatting guidelines and the HC Guidelines in the assignment instructions on Forum (near the top and bottom of the instructions respectively).**

Part 1: Variable Selection [#variables]

In this section, you will select, define, and examine a variable to measure in your city or town. To narrow the scope, you may select a sub-region in your city or town, such as a neighborhood or a particular establishment. If you are unfamiliar with the region, start by visiting and exploring for 30 minutes or so to help you select a variable. Above all, use common sense in choosing a location: obey local laws and do not put yourself in danger. Your instructor will understand if extreme circumstances (e.g., avoiding a disease outbreak) require you to modify this assignment to measure a variable accessible from the window of your room. However, they will also reward you for being creative and original. Read all notes below before selecting your variable.

Important notes and tips for the variable you select:

- This must be a variable in the sense that it varies somehow in your city. Perhaps it varies with respect to location or with respect to time.
- Given the above, the variable must be something for which you can make multiple measurements. For the purposes of this assignment, you must collect at least 10 distinct measurements, but the more the better to ensure that the statistical analysis to follow is sufficiently rich. Here are some examples:

  - If the variable varies spatially, you can measure the variable once on each block in a 10+ block neighborhood. For example, you could count the number of pieces of litter on each block. Note that a “city block” is usually defined as a region of a city surrounded on four sides by streets. You could also consider one “side” of the block (a street bounded by two intersections), but in this case, you should justify your choice of definition. Either way, be clear about how the blocks are defined.
  - You may be interested in measuring the cost of commonly purchased items. If so, you can measure the price of the item per establishment. For example, suppose that one neighborhood had 10+ different coffee shops. You could find the price of a medium sized cappuccino at each shop.
  - If the variable varies in time, you can measure the variable at different times of the day. For example, you could count the number of cars in a supermarket’s parking lot at 30 minute intervals for one day.
  - Another example of a quantity varying in time, right inside your own home, could be the height of the pile of dishes in the sink over different times of the day or days of the week.

- Get creative! Try to choose an interesting and informative variable.
- If you’re struggling to think of what to choose, or you don’t think that the above examples will work in your location, your professor would be happy to brainstorm with you in Office Hours. The success of your assignment relies on an appropriate variable choice.

1.1 Brainstorm variables

For this assignment, you must work with a quantitative variable so that you can calculate the mean, median, mode, and standard deviation. It may also be helpful to consider other types of variables that could be changing within your defined region or system. Give an example of the following variable types and explain why they fit the given classification. For consistency, you're recommended to consider variables that vary in the same way (spatially or temporarily) (<150 words total):

- a. Qualitative nominal
- b. Qualitative ordinal
- c. Quantitative discrete
- d. Quantitative continuous

- a. The race of people on the street. (I am not going to look at this variable; however, if I divide the data set by the person's race: Black, Asian, White, Hispanic, etc. these variables label the data without orders)
- b. Height of people but refer as short, normal, and tall. (I am also not going to measure this but if I set a standard: people who are up to 5ft tall are short, between 5 to 6ft are normal, and above 6ft is tall, these variables are ordinal variables show the orders.)
- c. The number of people who are wearing their masks incorrectly. (Humans can only be counted with integers. We cannot have a decmial number of people. They are counted with discrete units)
- d. The time when I measure actual data, percentage of people who are wearing their masks incorrectly (Time is a continuous variable since the concept of time is continuous. Even between minutes, we have seconds. Between seconds, we have microseconds and so on. The percentage is a calculated value; however, this is a quantitative continuous variable. There's no rule such as these variables should increase by a certain gap but can have any values between 0 to 100.)

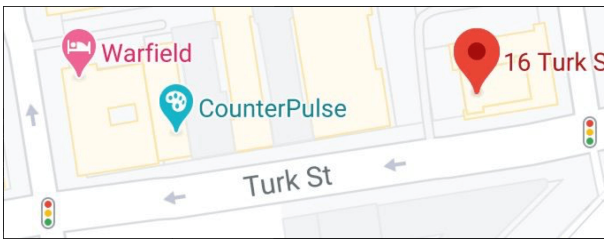
1.2 Select and describe your variable

Select and describe your variable your chosen quantitative variable here, being sure to address all items in this paragraph. First, describe why you selected this variable to focus on. Specifically identify whether this variable is continuous or discrete and explain why. Additionally, explain whether you will be measuring a total, proportion, or average, and the appropriate units for the corresponding measurement. Go further to explain in sufficient detail how exactly you will measure the variable, including the exact address(es) and time(s). Make sure that your explanation is clear enough that another student would understand how to make the same measurement. To aid in your explanation, provide an image that clearly identifies the region on a map and, if necessary, clarifies how the blocks are defined. (<150 words)

In [56]:

```
from IPython.display import Image
Image("Block map.png", height=300, width=300)
```

Out[56]:



I chose the percentage of people wearing masks incorrectly on the block in front of the res hall. Global pandemic threats all the Minervans' health, and it is essential that every Minervan wears a mask when they go outside. This variable is a good reminder for Minervans to wear their masks when they go out to protect themselves. This is proportional data calculated as the number of people on the block who are wearing masks incorrectly over the total number of people on the block. This is a quantitative continuous variable since the percentage variables don't show any relationship, such as increase/decrease by 1. Depends on the calculation, it could have some decimal points. I measured this variable for every 30 minutes from 9 AM to 6:30 PM, which is a time that I believe most Minervans go outside.

Part 2: Estimation and Measurement [#estimation]

Important note If there is any reason to believe that you did not authentically complete the **location based** portion of this assignment, you will be referred to the ASC. Please follow the instructions here carefully and include the original photo files in the zip folder along with the ipynb.

2.1 Go to a cafe in the neighborhood of your choice to produce a Fermi estimate of your variable.

For an authentic math-wiz experience in the city, go to a cafe in the neighborhood of your choice to produce a Fermi estimate of your variable using the back of a napkin from the cafe. It should be a true “back-of-the-napkin” estimation, so you may not (yet) make any measurements. If you’re unable to visit a cafe, simply grab a napkin from wherever you are to jot down your estimation. Scraps of paper or sticky notes would also work.

Estimation guidelines: *Your estimate should aim to involve at least 3 steps where you compute intermediate values. This means that there should be at least 3 different quantities computed along the way to the final quantity that you are estimating. Note that a calculation for the approximate geometric mean of a quantity, using upper and lower bounds, would be considered one step because that is only one quantity, even though it can be broken down further. Your initial napkin estimation can be quite rough, but note that when you type up your estimation (see below), you will have to describe each step clearly, show your work, state any assumptions you’re making, and discuss whether your answer seems plausible.*

2.2 Take some photos to document this experience. You must include:

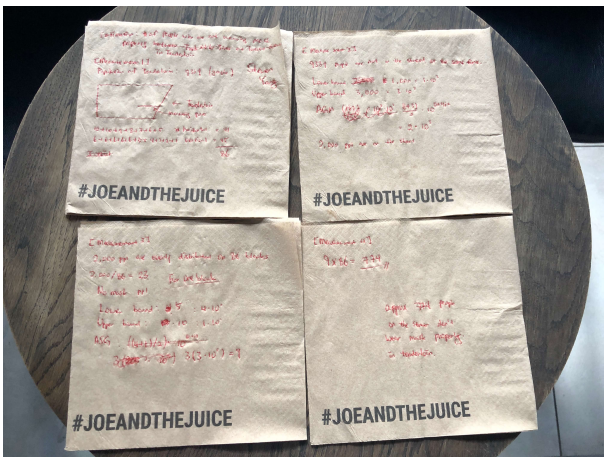
1. A photo of your “back-of-the-napkin” estimate (it can and should be quite rough and scribbly at this point). You will properly format the calculation later.
2. Two selfies at the cafe in which you constructed your Fermi estimate. Take one from inside, clearly show your face, your Fermi estimate, and some of the interior of the cafe. Take another one outside of the cafe showing your face and the exterior of the cafe, including the name. Bonus points if you are also holding your completed Fermi estimate in the photo. If you’re unable to visit a cafe for this assignment, simply include one selfie of you with your Fermi estimate, in the location at which you completed it.

In [57]:

```
# In the following cells, replace the filler image with your image
# Ensure the images are in the right folder location, then run the cell
# Add more cells if you have more photos

from IPython.display import Image
Image("Original Fermi.jpeg", height=300, width=300) # replace FILE_NAME_2.2_1.pn
g with your image file
```

Out[57]:

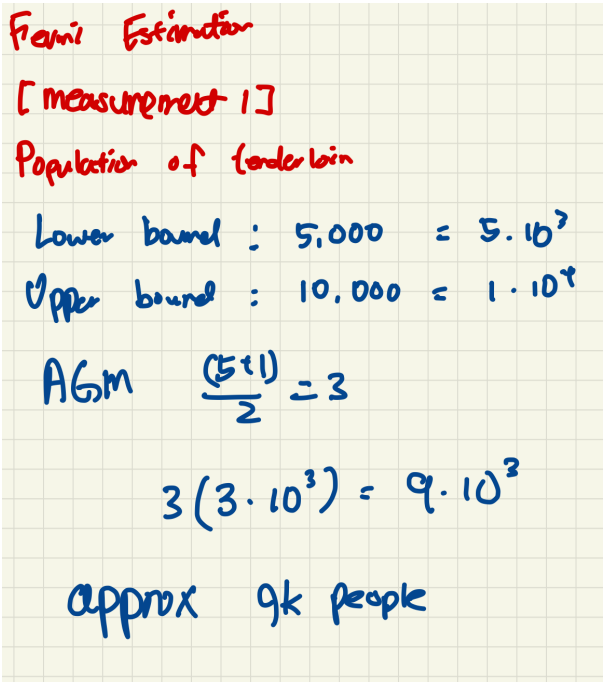


This is an original fermi estimation that I did at the cafe. However, Prof. Watson recommended me not to use Google at all for this assignment. Instead, I attach the following fermi estimations below, which I did in the res hall.

In [58]:

```
Image("Fermi1.png", height=300, width=300)
```

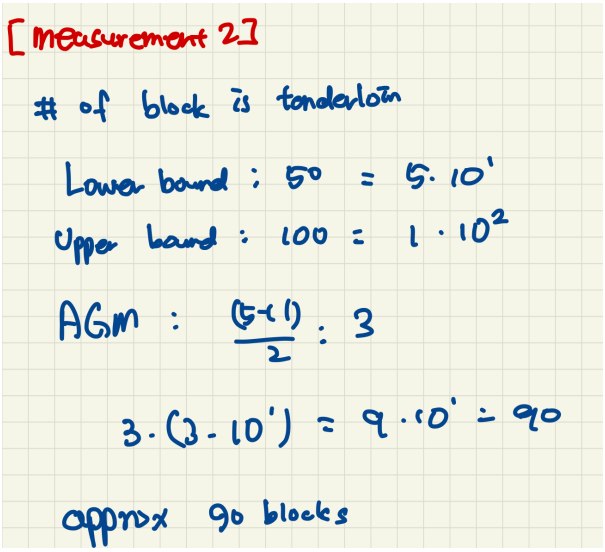
Out[58]:



In [59]:

```
Image("Fermi2.png", height=300, width=300)
```

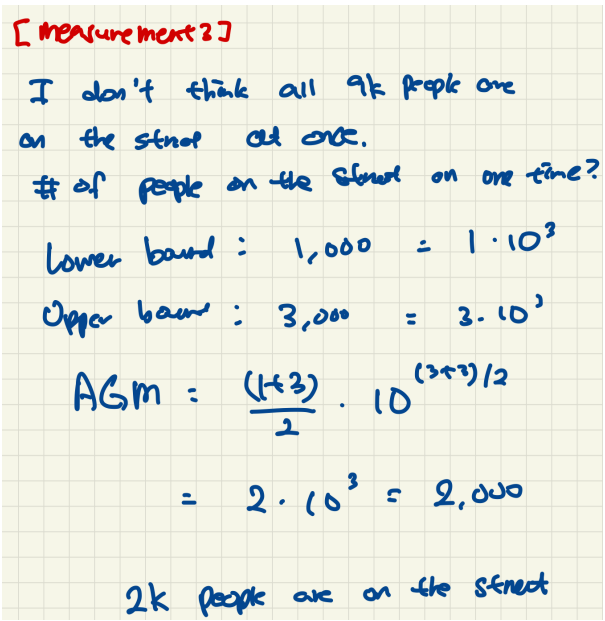
Out[59]:



In [60]:

```
Image("Fermi3.png", height=300, width=300)
```

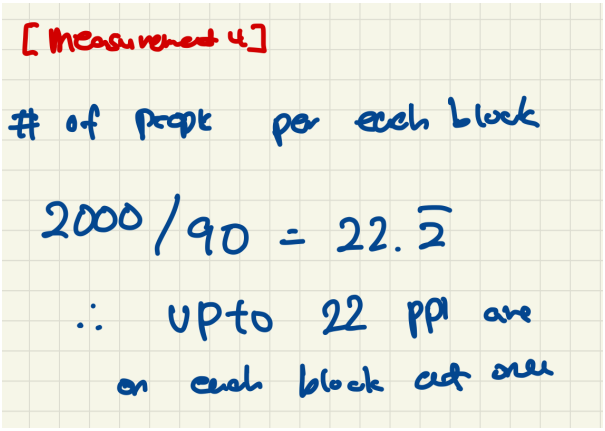
Out[60]:



In [61]:

```
Image("Fermi4.png", height=300, width=300)
```

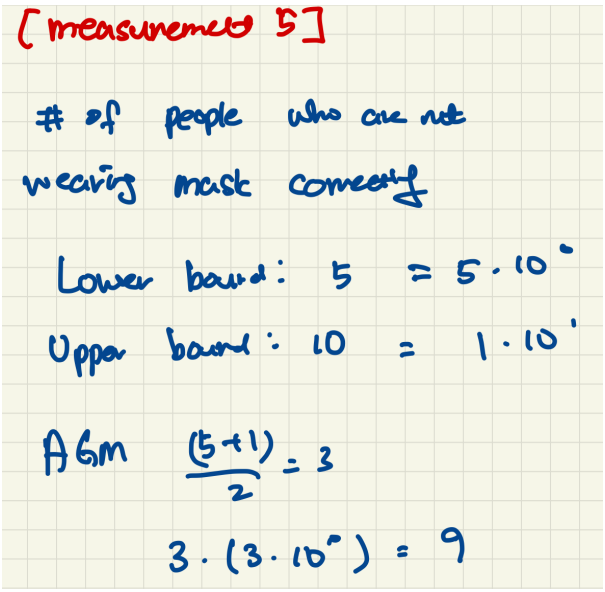
Out[61]:



In [62]:

```
Image("Fermi5.png", height=300, width=300)
```

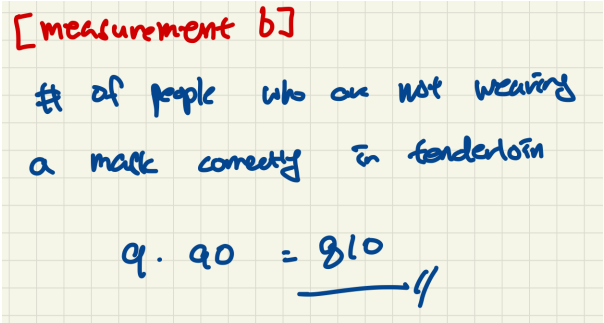
Out[62]:



In [63]:

```
Image("Fermi6.png", height=300, width=300)
```

Out[63]:



2.3 Typeset your full estimation in the Python notebook.

Here, be sure to clearly explain all steps, justify all assumptions, and comment on whether the answer seems plausible.

[Measurement 1]

Based on my observation during my life in Tenderloin, there were many stores on the street, and could not see many living places. I assume Tenderloin is not a big neighborhood and people normally live outside of the Tenderloin.

Lower bound: 5,000 = 5 *10^3 Upper bound: 10,000 = 1* 10^4

AGM: (5+1)/2 = 3, (3 *10^3) Therefore, the AGM is 9* 10^3 = 9,000.

There are approximately 9,000 people live in Tenderloin.

[Measurement 2]

There are a lot of blocks in Tenderloin. However, I observed some small blocks do not go all the way to Tenderloin's other side. I think counting big blocks of Tenderloins will better approximate the number of people wearing masks incorrectly per each block because small blocks are not significant.

Lower bound: 50 = 5 *10^1 Upper bound: 100 = 1* 10^2 AGM: (5+1)/2 = 3, (3 *10^1) Therefore, the AGM is 9* 10^1 = 90.

There are approximately 90 blocks in Tenderloin.

[Measurement 3]

It is not plausible to say all 9,000 people are on the street at the same time. Due to the global pandemic, people are not likely to come out. However, I understand that there are many homeless people in Tenderloin; I believe there are still some people on the street.

Lower bound: 1,000 = 1 *10^3 Upper bound: 3,000 = 3* 10^3 AGM = ((1+3)/2) *(10 ^ (3+3)/2) = 2* 10^3 = 2,000.

There are approximately 2,000 people are on the street at the same time throughout entire Tenderloin.

[Measurement 4]

I assume that the number of people on the street is fairly distributed on each block of Tenderloin. By simply dividing 2,000 by 90, I can get the approximate number of people on the street per block, which helps me set the lower/upper bound number of people who are wearing masks incorrectly.

2,000 / 90 = 22.2

There are approximately 22 people are on each block at once.

[Measurement 5]

Given that there are many homeless populations distributed in Tenderloin, I expect that people generally cannot afford their masks. My upper bound cannot exceed 22, which is the total number of people on each block, as I estimated.

Lower bound: 5 = 5 *10^0 Upper bound: 10 = 1* 10^1 AGM: (5+1)/2 = 3, (3 *10^0) Therefore, the AGM is 9* 10^0 = 9.

There are approximately 9 people who are wearing their masks incorrectly per each block.

[Measurement 6]

This calculation is based on the assumption: people are fairly distributed on each block throughout Tenderloin.

The number of people who are wearing their masks incorrectly per each block * The number of blocks

9 * 90 = 810.

There are 810 people who are wearing their masks incorrectly throughout Tenderloin at once.

2.4 It's time to collect your data!

Once again, take some photos to document your experience. Include at least two photos of your variable collection process. At least one photo should include your face and the variable you are counting.

In [64]:

```
Image("block_morning_1.jpeg", height=300, width=300)
```

Out[64]:

In [65]:

```
Image("block_selfie_morning_1.jpeg", height=300, width=300)
```

Out[65]:

These two photos are evidence for my measuring during the morning time.

In [66]:

```
Image("block1.jpeg", height=300, width=300)
```

Out[66]:

In [67]:

```
Image("block2.jpeg", height=300, width=300)
```

Out[67]:

In [68]:

```
Image("block3.jpeg", height=300, width=300)
```

Out[68]:

In [69]:

```
Image("block_selfie_night_1.jpeg", height=300, width=300)
```

Out[69]:

These four photos are evidence for my measuring during the evening time.

In [70]:

```
Image("robert.png", height=100, width=200)
```

Out[70]:

ALERT: Due to my class schedule, I asked Robert D. for collecting four of my data points: 5 PM, 5:30 PM, 6 PM, and 6:30 PM on Nov 3rd. Since a different observer collected those four data points, there might be some little different observation standards, which possibly affects the data's accuracy. To minimize this error, I provided a guideline below for Robert.

In [71]:

```
Image("correct_mask.png", height=300, width=300)
```

Out[71]:

Part 3: Analysis [#algorithms, #dataviz, #descriptivestats]

3.1 Analyze the data in Python

a. Import the data

Use any method to import your collected data into Python. You can simply type the data directly into a Python list or numpy array. Or, you can put the data in a Google sheet, export to a .csv file, and import into Python. Print your data here.

In [72]:

```
# Edit this cell to answer 3.1.
#Import packages
import pandas as pd #Use pandas to import .csv file
pd.set_option('max_rows', 100) #shows all parts of table from the data set
#tells matplotlib to display figures in the notebook and not in a separate windo
w
import seaborn as sns #use seaborn package for in case if I want to make my #dat
aviz look nicer
sns.set(color_codes=True)

#import data from simplified .csv file
data = pd.read_csv("Assignment 2 Data collection - Sheet1.csv")
time = data["Time"]
percentage = data["% of people who are wearing their masks incorrectly"].tolist
()#define list for the future calculations
data
```

Out[72]:

	Time	# of people who are wearing their masks incorrectly	Total # of ppl on the block	% of people who are wearing their masks incorrectly
0	9 AM Nov 1	6	16	37.5
1	9:30 AM Nov 1	7	13	53.8
2	10: 00 AM Nov 1	13	27	48.1
3	10:30 AM Nov 1	15	25	60.0
4	11 AM Nov 1	7	20	35.0
5	11:30 AM Nov 1	11	25	44.0
6	12 PM Nov 1	13	32	40.6
7	12:30 PM Nov 1	5	27	18.5
8	1:00 PM Nov 1	16	23	69.6
9	1:30 PM Nov 1	9	15	60.0
10	2:00 PM Nov 1	0	10	0.0
11	2:30 PM Nov 1	8	27	29.6
12	3:00 PM Nov 1	9	25	36.0
13	3:30 PM Nov 1	0	7	0.0
14	4:00 PM Nov 1	13	26	50.0
15	4:30 PM Nov 1	1	15	6.7
16	5:00 PM Nov 1	4	10	40.0
17	5:30 PM Nov 1	3	12	25.0
18	6:00 PM Nov 1	1	3	33.3
19	6:30 PM Nov 1	3	11	27.3
20	9 AM Nov 3	7	17	41.2
21	9:30 AM Nov 3	5	12	41.7

	Time	# of people who are wearing their masks incorrectly	Total # of ppl on the block	% of people who are wearing their masks incorrectly
22	10: 00 AM Nov 3	15	26	57.7
23	10:30 AM Nov 3	15	23	65.2
24	11 AM Nov 4	9	21	42.9
25	11:30 AM Nov 4	7	20	35.0
26	12 PM Nov 4	11	30	36.7
27	12:30 PM Nov 3	3	25	12.0
28	1:00 PM Nov 3	14	22	63.6
29	1:30 PM Nov 3	7	14	50.0
30	2:00 PM Nov 3	1	14	7.1
31	2:30 PM Nov 3	10	30	33.3
32	3:00 PM Nov 3	7	24	29.2
33	3:30 PM Nov 3	1	24	4.2
34	4:00 PM Nov 3	11	25	44.0
35	4:30 PM Nov 3	1	12	8.3
36	5:00 PM Nov 3	5	9	55.6
37	5:30 PM Nov 3	10	15	66.7
38	6:00 PM Nov 3	10	12	83.3
39	6:30 PM Nov 3	3	8	37.5

b. Compute descriptive statistics

Using Python, calculate the mean, median, mode, range, and standard deviation of your variable, and print these values. Your code should not rely on library functions or packages directly, meaning that you cannot simply call `np.mean(data)`, `stats.mean(data)`, or other similar functions. Instead, write your stats calculators "*from scratch*"! This approach, along with sufficient in-line comments to explain the code, will allow you to demonstrate a strong understanding of #descriptivestats. If you use any built-in or imported functions within one of the steps of the function, (to sum or sort the data, for example) you need to explain how it works. Do not blindly use library functions without reviewing the documentation!

In [73]:

```
#define mean function
def mean(lst):
    mean = sum(lst) / len(lst)
    return mean

#median calculation
def median(lst):
    lst.sort()
    if len(lst) % 2 == 0: # Finding the position of the median if the # of eleme
nts is even
        first_median = lst[len(lst) // 2]
        second_median = lst[len(lst) // 2 - 1]
        median = (first_median + second_median) / 2
    else: # Finding the position of the median if the # of elements is odd
        median = lst[len(lst) // 2]
    return median

#mode calculation
def mode(lst):
    L1=[] #count the occurrence of each number and append or add findings to the
list

    i = 0 #count the numbers and put them into L1
    while i < len(lst) :
        L1.append(lst.count(lst[i]))
        i += 1

    # the occurrences for each number in sorted lst
    # create a custom dictionary d1 for k : V
    # k = value, v = occurence

    d1 = dict(zip(lst, L1))
    d2={k for (k,v) in d1.items() if v == max(L1)} # the k values with the highe
st v values.
    return str(d2)

#range calculation
def rng(lst):
    return max(lst)-min(lst)#range is equal to maximum vlaue - minimum value

#standard deviation calculation
def standard_dev(lst):
    variance = sum([((x - mean(lst)) ** 2) for x in lst]) / len(lst) #interpret
 the formula for standard deviation
    sd = variance ** 0.5 #sqrt
    return sd

print("Mean is:", round(mean(percentage),2))
print("Median is:", median(percentage))
print("Modes are:", mode(percentage))
print("Range is:", rng(percentage))
print("Standard Deviation is:", round(standard_dev(percentage),2))
```

```
Mean is: 38.25
Median is: 38.75
Modes are: {0.0, 33.3, 35.0, 37.5, 44.0, 50.0, 60.0}
Range is: 83.3
Standard Deviation is: 20.07
```

c. Histogram

Create a histogram for your data, properly formatting your figure.

In [74]:

```
import matplotlib.pyplot as plt #importing matplotlib package
%matplotlib inline
#histogram coding
plt.hist(percentage, bins = [0,10,20,30,40,50,60,70,80,90,100], color = "r", alp
ha=1, linewidth=5)
plt.title("Distribution of percentage of people who are wearing their masks inco
rrectly.", fontsize = 15)
plt.xlabel("Percentage (%)", fontsize = 10)
plt.ylabel("Frequency of data", fontsize = 10)
txt="Fig. 1. 40 Data points are collected in this histogram.\n There are 10 bins
in this histgoram and all data points are distributd in each bin.\nThis histogra
m is slightly skewed to the left."
plt.text(45,-3,txt, horizontalalignment = "center")
plt.xticks([0,10,20,30,40,50,60,70,80,90,100]) #increase xticks by 10
plt.yticks([0,1,2,3,4,5,6,7,8,9]) #increase yticks by 1
plt.axvline(mean(percentage), color='g', label = "Mean = 38.25") #label legend f
or mean
plt.axvline(median(percentage), color='y', label = "Median = 38.75")#label legen
d for median
plt.legend()
ax=plt.gca()
ax.set_facecolor('w')
plt.show()
```

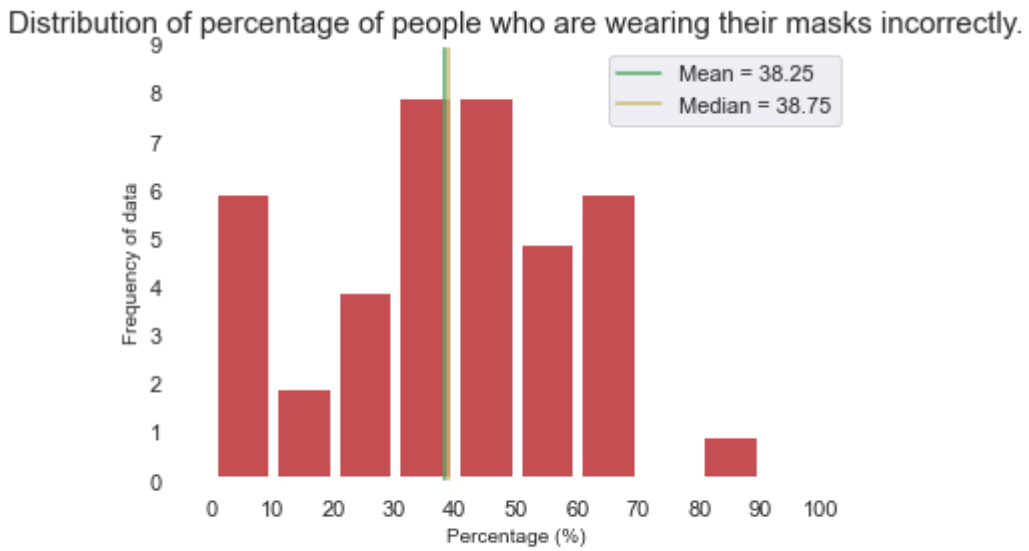


Fig. 1. 40 Data points are collected in this histogram.
There are 10 bins in this histgoram and all data points are distributd in each bin.
This histogram is slightly skewed to the left.

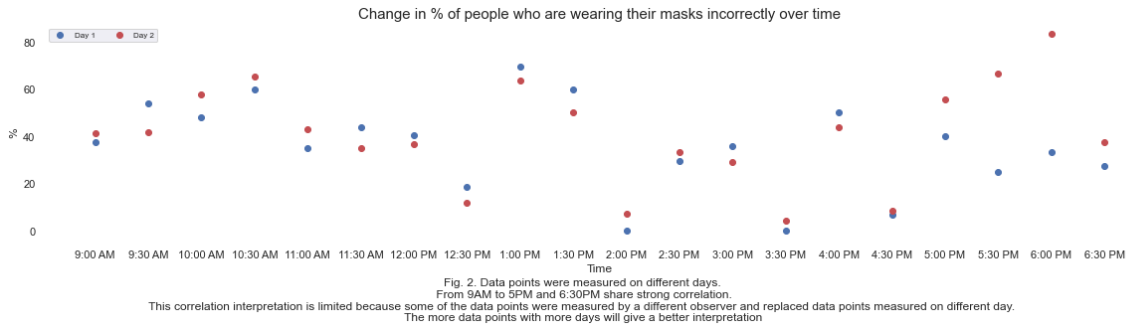
d. Scatterplot (Steven's Voluntary)

In [75]:

```
data = pd.read_csv("Assignment 2 Data collection - Sheet2.csv") #import another
 spreadsheet for scatterplot
time = data["Time"]
day1 = data["% Day 1"]
day2 = data["% Day 2"]

#scatter plot coding
d1 = plt.scatter(time, day1, color = "b")
d2 = plt.scatter(time, day2, color = "r")
plt.legend((d1,d2),('Day 1', 'Day 2'),scatterpoints=1, loc='upper left', ncol=3,
fontsize=8)


plt.title("Change in % of people who are wearing their masks incorrectly over ti
me", fontsize = 15)
plt.ylabel("%")
plt.xlabel("Time")
txt="Fig. 2. Data points were measured on different days.\n From 9AM to 5PM and
 6:30PM share strong correlation.\n This correlation interpretation is limited b
ecause some of the data points were measured by a different observer and replace
d data points measured on different day. \n The more data points with more days
 will give a better interpretation"
plt.figtext(0.5, -0.15, txt, wrap=True, horizontalalignment='center', fontsize=1
2)
fig = plt.gcf()
fig.set_figwidth(20)#increase the width to see a full scatterplot clearly
ax=plt.gca()
ax.set_facecolor('w')
plt.show()
```



3.2 Interpret the results

What can you say about the neighborhood based on each of these descriptive statistics and the visualization? In your response, comment on the shape of the histogram and whether it is in agreement with the set of descriptive statistics. (<200 words)

The histogram shows that there are lots of data-focused around 40. However, the standard deviation is 20, which indicates it has a wide range of distribution between 20 to 60. This means the percentage of people wearing masks incorrectly varies in a large range. It's much symmetric on the histogram from 20 to 60; therefore, we can say around 40% of people are wearing their masks incorrectly on the block.

The histogram is slightly skewed to the left because the mean value is smaller than the median value. There are seven mode values. Five of them are between 30 to 50 and I can see the graph shows the highest frequency between 30 to 50. There's one outlier between 80% to 90%. Sometimes, most people wear masks incorrectly shown by the outlier far right in the histogram.

Additionally, I have a scatterplot since I wanted to see a correlation of percentages over time. I only measured two days, which means there are only two big data sets. Even though there are only two data sets, it seems 9 AM to 5 PM and 6:30 PM have similar percentages. Understand that a different observer collected four data points, and I only have two big data sets; there are possible errors in these data sets. In the future, if I collect more data sets and attach them to this scatter plot, it is more likely to show a stronger correlation between the percentage and time so that I can observe how the percentage changes over time.

Part 4: Probability Considerations [#probability, #algorithms]

Suppose something unfortunate happened: you grabbed too many napkins for your Fermi estimate, so you decided to write all of your variable measurements on separate napkins, one napkin for each measurement. You had them all in order, from the first measurement to the last, but on your way back to the residence (or home), the wind picked up and blew them all away! Luckily, you managed to collect all of the napkins, but now the data is totally randomly reordered, meaning that you have no idea about the order in which the measurements were taken. Too bad you didn’t label them! Suppose that you tried to just guess the correct napkin order randomly. In other words, you randomly assign each napkin to a given spot (first measurement collected, second measurement collected, third, … etc).

4.1 Matching Napkins Simulation

What is the probability that you are unlucky, and sadly NONE of the napkins are matched to the correct spot (you guessed all of them wrong)? **Write a simulation in Python to estimate this probability**. Ensure that your code has sufficiently detailed comments to explain the key steps in the solution. *Use your own data*. If there are repeated values in your measurements, take that into consideration: what if two napkins for two distinct measurements show the same value, and are then swapped? Would those napkins be considered a mismatch, or a lucky match? Take note of any peculiarities in your values. Hints to get started (use these if you wish, keeping in mind that there are multiple approaches):

- You can define your stack of napkins with measurements in the correct order as a list in Python. For example, you can set `napkins = [measurement1,measurement2,...,measurement10]`. This means that napkins[0] should give the first measurement.
- A random permutation of this list can be created with the following code: `rand_napkins = np.random.choice(napkins,10,replace=False)`. You should be able to explain how this function works, including the parameters passed, and why it is relevant for the problem. Read the Numpy documentation (https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html) to learn more.
- If you have 10 measurements, you will want to check whether `rand_napkins[i] == napkins[i]`, for each value of `i` from 0 to 9.
- You'll need to use a loop to create many random lists and repeat the checking procedure, keeping track of the number of matches each time.
- More Python tips at the end of the assignment instructions in Forum.

In [76]:

```
import random

napkins = percentage
randomizer = percentage.copy() #copy the list


totalnomatches = 0
for i in range(10000): #10,000 rounds of simulation!
    count_dismatch = 0
    random.shuffle(randomizer) #shuffle the copied list
    for j in range(len(napkins)):
        if napkins[j] != randomizer[j]: #when the original set is not matched wi
th shuffled set
            count_dismatch += 1
    if count_dismatch == len(napkins):
        totalnomatches += 1
#if two napkins for two distinct measurements show the same value, and are then
 swapped, they are lucky match.

print((totalnomatches / 10000) * 100,"%")
```

```
25.040000000000003 %
```

4.2 Fully analyze your simulation

Fully analyze your simulation by addressing the following points (<250 words):

- How does your simulation work as a whole?
- How does the simulation approach differ from calculating the probability analytically? You need not perform the analytical calculation (see the last optional challenge below).
- Interpret the result using the appropriate interpretation of probability.

This simulation counts lucky match as count. Since I am only looking at 40 data points, which are just numbers of percentages, I thought even I had a lucky match; it should not be a big problem. If I didn't count lucky match as count, the probability from the simulation would be lower.

First of all, I made a list of percentages and copied this list so that I can see whether the original list and copied list match or not. There are two for loops: one for 10,000 rounds of simulations and one for one comparison for the list for each round. Inside the second loop, the original list and shuffled copied list keep comparing the values until they figure out the mismatch. If there's one mismatch, the total mismatch gets to count. This process is being repeated 10,000 times. The total number of mismatches is divided by 10,000, which is the total number of the simulation to get the simulated possibility.

In an analytical approach, the analysis relies on theoretical value. The theoretical value is more reliable. However, theoretical probability tells us a basic idea of possibility in reality. The simulation is the value that we observe in reality but changes every time we observe certain cases. This simulated probability gets closer to the theoretical value as we repeat the trials. I have 10,000 times for doing a simulation in my code, and I continuously observe around 25%. Therefore, I can expect that analytical probability is also approximately 25% even I have not to proceed with any actual calculation.

4.3 *Optional*: Expected Value

What is the expected number of napkins that will be correctly matched to the corresponding block? Estimate this probability using a simulation and provide a detailed interpretation of the result.

In [77]:

```
import random

def EV(n):
    napkins = percentage
    randomizer = percentage.copy()
    total_match=0 #store total # of matches
    for i in range(n):
        napkins = percentage
        random.shuffle(randomizer)
        match_count = 0
        for j in range(len(napkins)):
            if napkins[j] == randomizer[j]:
                match_count += 1
        total_match += match_count

    return (total_match/n)

EV(10000)
```

Out[77]:

```
1.3711
```

Like the simulation over, it continuously checks whether the original sets and randomized set match each other. In every single round, if there's any napkin matches each other, it gets counted. To process this, I have match_count += 1 in my code. The total match number gets added as much as counted as match_count from each round at every end of the round. This total number of matches will be divided by n, the number of repeated simulations. Therefore we get the expected value.

4.4 *Optional*: Probability Distribution

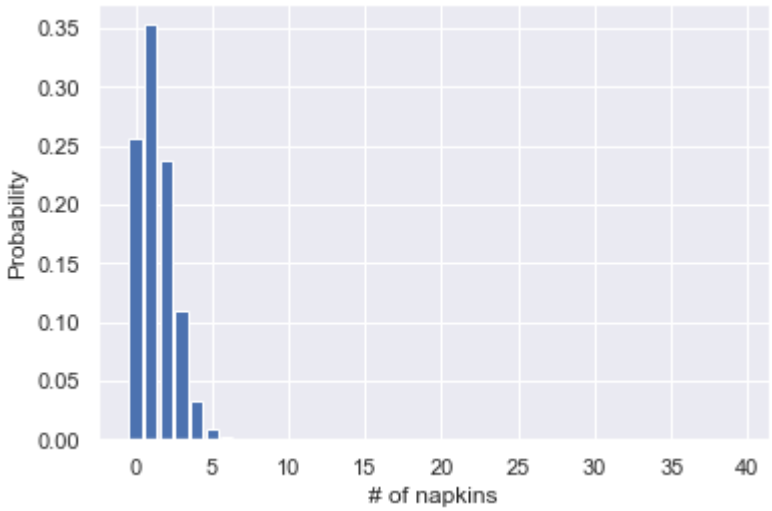
Determine the probability distribution as a function of the number of correctly matched napkins and create a visualization. Provide a detailed interpretation of the result.

In [78]:

```
import matplotlib.pyplot as plt

def distribution(n):
    total_match = []
    napkins = percentage
    randomizer = percentage.copy()
    for i in range(n):
        napkins = percentage
        random.shuffle(randomizer)
        match_count = 0
        for j in range(len(napkins)):
            if napkins[j] == randomizer[j]:
                match_count += 1
        total_match.append(match_count)
    prob = []
    for k in range(len(napkins)):
        freq = total_match.count(k)
        prob.append(freq/n)
    plt.bar(range(len(prob)),prob)
    plt.ylabel("Probability")
    plt.xlabel("# of napkins")
    plt.show()

distribution(10000)
```



I adapted the structure of the code in the previous cell. However, some additions here: opened list for the probability, and every match count's probability is kept recorded in the opened cell. After this repeated recording, another for loop calculates the probability based on the frequency.

After that, plt draws a bar chart to show how the probability is distributed.

4.5 *Optional*: Analytical Computation

Compute the probability or expected value found above, or both, analytically (without a simulation). For a strong score, this must involve a full derivation rather than a one-line answer. Provide a thorough explanation for your approach, enough so that your peers could understand the solution using concepts from class alone. If any mathematics or statistics concepts are used that are beyond the scope of class, you must include an APA citation for the source of the material, as well as a description of its relevance for the problem.

In [79]:

```
Image("4.5excel.png", height=300, width=300)
```

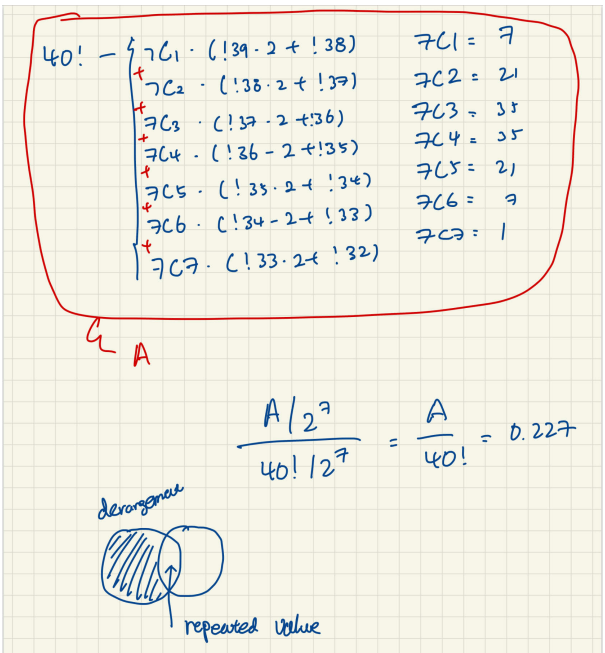
Out[79]:

	A	B	C	D	E	F
1	40!	8.16E+47				
2	!40	3.00E+47				
3	!39	7.50E+45	7C1	7	raw1	1.06E+47
4	!38	1.92E+44	7C2	21	raw2	8.19E+45
5	!37	5.06E+42	7C3	35	raw3	3.59E+44
6	!36	1.37E+41	7C4	35	raw4	9.71E+42
7	!35	3.80E+39	7C5	21	raw5	1.62E+41
8	!34	1.09E+38	7C6	7	raw6	1.54E+39
9	!33	3.19E+36	7C7	1	raw7	6.49E+36
10	!32	9.68E+34			SUM	1.15E+47
11						
12					!40-SUM	1.85E+47
13	2^7	128				
14	40!/(2^7)	6.37E+45		Answer(!40-SUM)/40	2.27E-01	

In [80]:

```
Image("4.5hand.jpeg", height=300, width=300)
```

Out[80]:



Understood the concept of the derangement. Followed the problem solving step from the video. Calculation has made on excel.

blackpenredpen. (2018, November 17). YouTube [Video]. YouTube. https://www.youtube.com/watch?v=ZT8zBvCr7L8&feature=youtu.be (https://www.youtube.com/watch?v=ZT8zBvCr7L8&feature=youtu.be)

Part 5: Reflection

Reflect on what you learned about the HCs in this assignment, focussing on the connections between the HCs, and their connections to the region. Also reflect on how your prediction and estimation from parts 1 and 2 compare to the results. (<200 words)

By applying the Fermi estimation to my region, I understood the difference between variables and chose a quantitative discrete variable: the number of people wearing their masks incorrectly. Fermi estimation has been processed based on my local experiences.

After collecting data, I put them into descriptive stats to see five values: mean, median, mode, range, and standard deviation. I understood that data points are widely distributed based on SD, which is 20. However, since the mean and median share similar values, I expected that data are symmetrically distributed. After displaying the data using data visualization, the histogram showed not perfect but an acceptable symmetric distributed shape. Additionally, I did a scatter plot to see how data from other days are correlated. Data points were close enough from 9 AM to 5 PM and 6:30 PM. Robert measured the other day's data from 5 PM to 6:30 PM. The data have some differences because there were two different observers. I provided the guideline to determine who's wearing the mask incorrectly. To see a stronger correlation and be more accurate, I can spend more days in the future to collect more data points.

To make sure my code is working and has no logical errors, I ran several times and got proof-reading from peer tutors. Also, to minimize my grammar mistakes, I put all the sentences into Grammarly and corrected them.

I expected 30 to 40% of people wear masks incorrectly and based on that expectation, and I set the estimation. I estimated 810 people are wearing their masks incorrectly among 2,000 people on the street from my fermi estimation. This means my estimation says 40.5% of people are wearing their masks incorrectly. It's a little over my prediction but still close enough to my prediction.

It would be better if I could explore the whole Tenderloin and count every block to see this relationship, but it requires a massive amount of time. This LBA assignment can be explored further.

You're done!

You must upload TWO files:

1. A **zipped folder** containing the .ipynb file, your image files, and any other relevant files for running the notebook.

1. A **PDF** of your entire assignment. A PDF of your entire assignment. This is to be submitted as a separate file, NOT simply inside the zipped folder. Email attachments will not be accepted. We encourage students to follow the tips available in this guide (https://docs.google.com/document/d/1gRMol9Ebbvyu1mvEKzma92o_N7ZbNXsPIb1QdQV0TeE/edit?usp=sharing), *especially the best practices listed at the end*.