**MAT 128 Lab 8: Estimating probabilities and Normal Distributions**

**Part A**

**Get the Data**

We will use the NYC Open Data 311 service request dataset:

https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

This is a record of all 311 service requests from 2010 to the present.

1) As in Lab 4, filter the data by the "Created Date" so that it is only one or two days. For example, filtering so that the "Created Date" is between 02/21/2018 12:00:00 AM and 02/23/2018 12:00:00 AM will give all 311 service requests created on Feb. 21 and 22, 2018.

2) Download the filtered data as a CSV file.

This lab will assume this CSV file is called 311_Service_Requests.csv and located in the same directory (folder) as your code.

**Explore the Data**

1) Write code to import the pandas library.

2) Read the 311 service request CSV file into a dataframe called `requests`.

3) Run your code and type `requests.head()` in the console to see that the CSV file was read correctly. We are going to focus on the `Complaint Type` column which describes the complaint, and the `Descriptor` column, which gives additional information about the complaint.

4) Count the number of complaints of each type and print them by adding this code to your program:

```
complaint_counts = requests['Complaint Type'].value_counts()
print(complaint_counts)
```

`complaint_counts` is a Pandas series that is indexed by the type of complaint.

5) We can also plot the number of each complain as a bar graph:

```
complaint_counts.plot.bar()
```

6) Since there are so many complaints, this graph is hard to read. To improve it, we will only plot the complaint types with more than 100 complaints.

First make a condition variable:

```
common = complaint_counts > 100
```

Then use this variable to only select the rows meeting its condition:

```
complaint_counts[common].plot.bar()
```

Here is another way to do this is one line, by not using a variable:

```
complaint_counts[complaint_counts > 100].plot.bar()
```

When we selected rows from our taxi dataframe (Lab 6), we needed to give the column the condition was based on. Since `complaint_counts` is a series, or a single column, we do not need to give any other information.

*Challenges:*

- Make a bar graph of all complaints with more than 500 counts
- Add a title to your plot
- Change the color of the bars to red

**Analyze the data**

We can estimate the probability of an event (eg. a 311 service request is about illegal parking) using this formula:

$$\text{Probability of an event} = \frac{\text{\# of times the event occurs}}{\text{Total \# of possible outcomes}}$$

So to estimate the probability that a 311 service request is about illegal parking, we can compute:

$$\text{Probability 311 request about illegal parking} = \frac{\text{\# requests about illegal parking}}{\text{Total \# of requests}}$$

We will now write code to compute this:

1) Create a condition variable called `illegal` to select all rows with `Complaint Type` of `Illegal Parking`
2) Count the number of rows meeting this condition:

```
num_illegal = len(requests[illegal])
```

3) Count the number of rows in the full dataframe

.

```
num_total = len(requests)
```

4) Estimate the probability and print it out:

```
prob = num_illegal/num_total
print("The estimated probability that a 311 request is about illegal parking is",prob)
```

*Challenge:* Two of the Descriptors for the Rodent complaint type are “Rat Sighting” and “Mouse Sighting”. What is the probability that a Rodent complaint is a “Rat Sighting”? What is the probability that a Rodent complain is a “Mouse Sighting”?