AM 170B HW1

Simon Lee

April 5, 2022

All the following plots from here and moving forward were generated on a jupyter notebook using python.

1 COVID Data Visualization

1.1 Regressions on cases vs deaths

In the first part of the assignment we downloaded the COVID cases and death counts directly from the CDC.gov website. The first plot we were asked to generate was a total number of cases vs. deaths per state graph. However in the field of data sciences generating a scatter plot on its own is not enough to interpret data. Therefore one of the most basic practices in machine learning is curve fitting. Curve fitting is the process of fitting a mathematical function that best represents the trends of our data. Using numpys poly1d method, we generate the two following graphs:

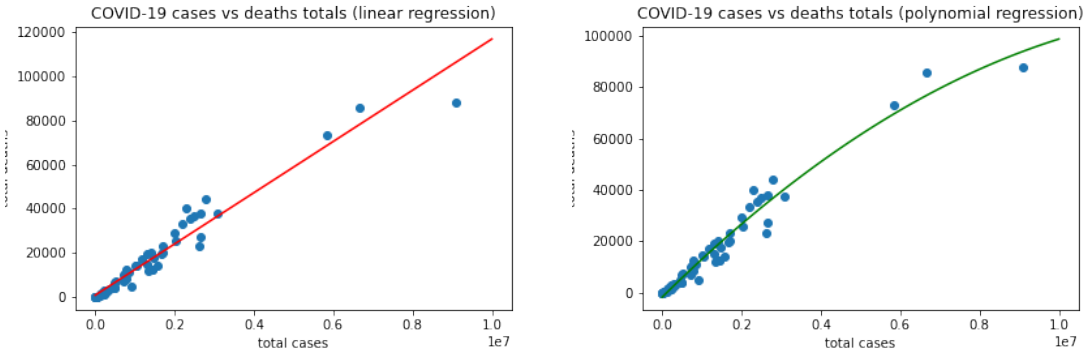


Figure 1: The cases vs. death graphs fitted with a linear regression (left), and polynomial regression (right)

The relationship I can make from these graphs is that the cases to death ratio is roughly 1:0.01. Most of the states on the graph have fewer than $0.4e7$ cases and we can see in the linear regression that most states are actually along that line. However we see that there are three states in which two are under fitted and one is over fitted. However the polynomial fit appears to be done very well and it better captures the data which logically makes sense as most data is best represented with more than a linear function.

1.2 Cases in Descending Order

Next we were asked to make a histogram of the cases in descending order. So we took the most recent day on our dataset, March 30th, 2022 and sorted the data frame to make this graph.

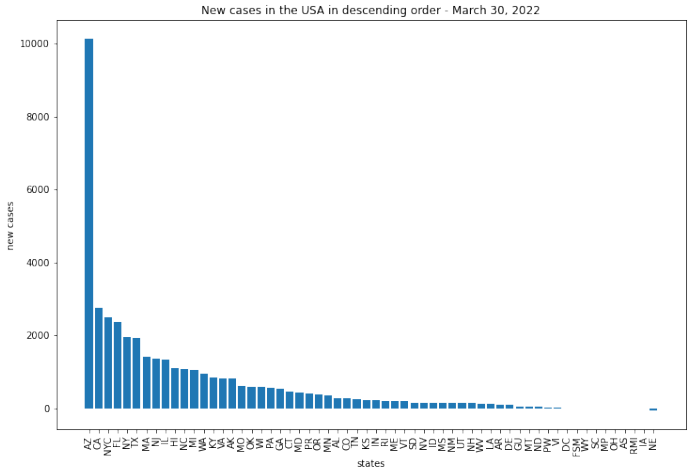


Figure 2: Cases by state in descending order

Interestingly there seems to be some error with the NE state because we can see there is some value but for whatever reason it is ranked last in the histogram plot. Also there appears to be more than 50 "states" into the dataset. Though I did not filter them out, I likely could have through more manipulation in the dataframe.

1.3 Time series of California, Washington, Georgia, Kentucky

The last part of the part I, was to plot the time series of several states. We were asked to plot the cases per day as well as the cases totals over a time domain. We generated the following:

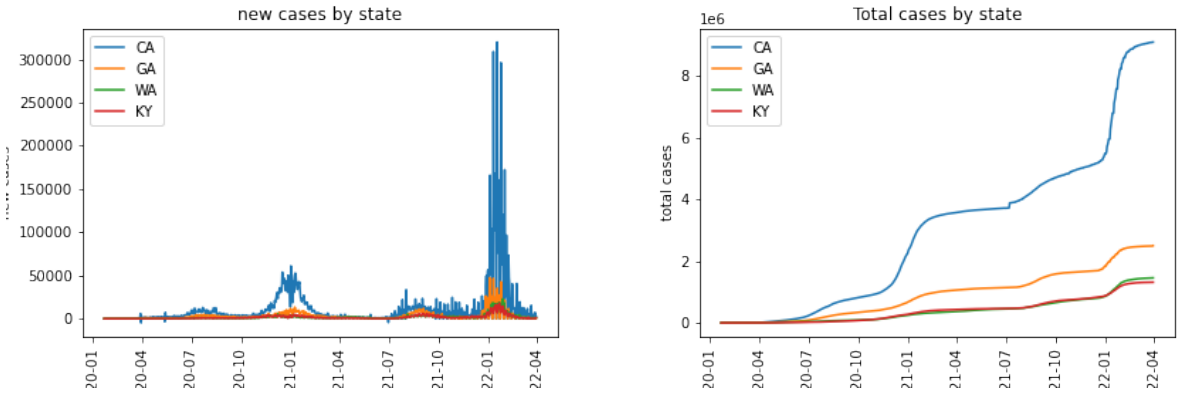


Figure 3: Time series of new cases per day (left) and case totals (right)

We also generated these same graphs by applying a normalization. The normalization I decided to use was the to rescale each state by 100,000. This simple normalization technique rescales the range of features to scale the range in [0, 100000]. Therefore the representation is widely different to the first set of graphs but the cases per day and the total cases are now within this range of [0, 100000]. The purpose of normalization is to transform data in a way that they are dimensionless. Below we show the populations in a table that helped us achieve this normalization:

States	Population (/millions)
CA	39.51
GA	10.62
KY	4.47
WA	7.62

Table 1: Table showing off states and their populations.

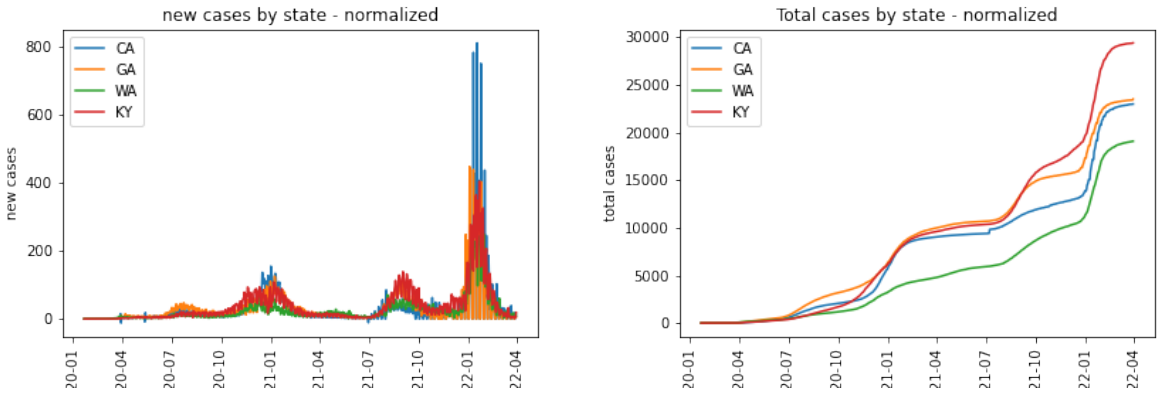


Figure 4: Time series of new cases per day (left) and case totals (right) - (Normalized with the min-max feature scaling)

In this normalized data we can see that California had the worst cases per day but Kentucky had the worse total case counts relative to their population. This normalization helps us see how different populations can show drastically different results and why normalization is a helpful technique in data science to make our data dimensionless.

2 II Gene expression differences between normal ovary tissue and ovarian cancer biopsies.

In the second part of the assignment we took ovarian cancer data and ran various statistical tests and graphed them.

2.1 cancer vs normal samples

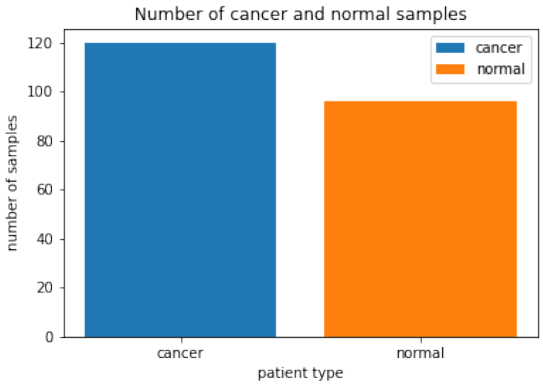


Figure 5: cancer vs normal counts

The first part of this problem was to count how many normal and cancer samples we had. We made a simple barplot as seen in Figure 5. Moving forward we have marked normal samples to be orange and cancer samples to be blue and it will be analogous in the next few graphs for continuity purposes.

2.2 plotting the samples + log2 fold

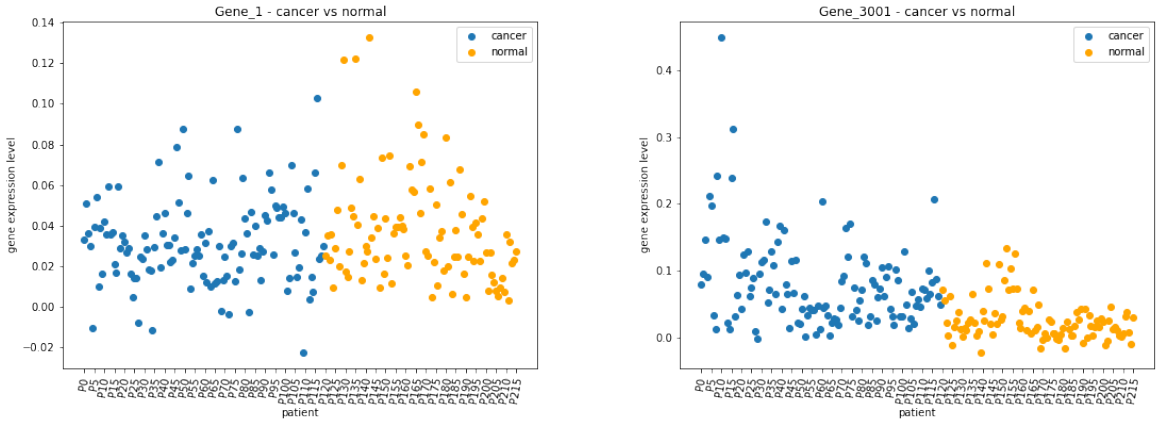


Figure 6: plotting samples: gene1 (left), gene 3002 (right)

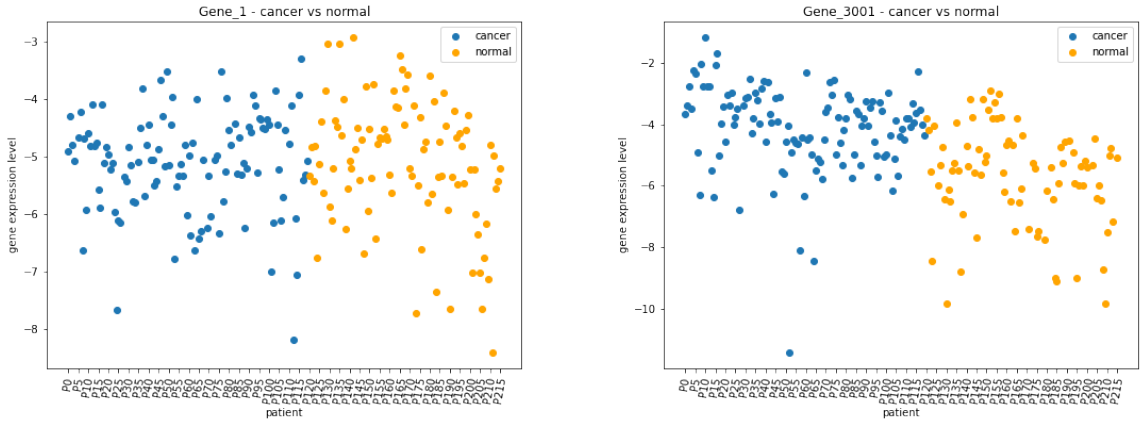


Figure 7: plotting log2 fold samples: gene1 (left), gene 3002 (right)

In these next four plots we plot the samples of gene_1 and gene_3002. In Figure 6 we plotted the gene expression levels against patients. I noticed an interesting observation after plotting these set of graphs. I noticed in Figure 6 some had negative gene expression and I was wondering what this mean or how it could be possible.

In Figure 7 we ran the log2 fold which is a statistical test in which the log-ratio of a gene's or a transcript's expression values in two different conditions. Therefore, we took advantage of Numpy, which has a log2 fold method and we used this to get our new graphical representation that you see in Figure 7. We can see that by applying a log2fold that gene_1 and gene_3002's gene expression values are now all negative.

2.3 T-test & Mann-Whitney-Wilcoxon tests

```
T-test value for gene_3002:  Ttest_indResult(statistic=-7.379612834453349, pvalue=3.4629258618496455e-12)
T-test value for gene_1:  Ttest_indResult(statistic=1.7501709816564117, pvalue=0.08152199430351342)
Mann-Whitney-Wilcoxon value for gene_3002:  MannwhitneyuResult(statistic=2149.0, pvalue=1.2821739807700903e-15)
Mann-Whitney-Wilcoxon value for gene_1:  MannwhitneyuResult(statistic=5373.0, pvalue=0.19855158370498954)
```

Figure 8: t-test and Mann-Whitney-Wilcoxon values

Lastly we ran some statistical tests on our data and printed them out to our machine. The two test we ran for this portion of the problem were the T-test & Mann-Whitney-Wilcoxon tests. A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features. We used the ttest_ind method from scipy.stats to calculate this test. We also have the Mann-Whitney-Wilcoxon tests, which is a method that wants to see if the two populations have the same shape. scipy.stats yet again contains a mannwhitneyu method that we used to calculate these numbers. The results are displayed in Figure 8.

3 III Population dynamics: predator prey.

Lastly we have the predator prey model, which is a type of dynamical systems model that simulates the population of a specific predator and prey over a range of time.

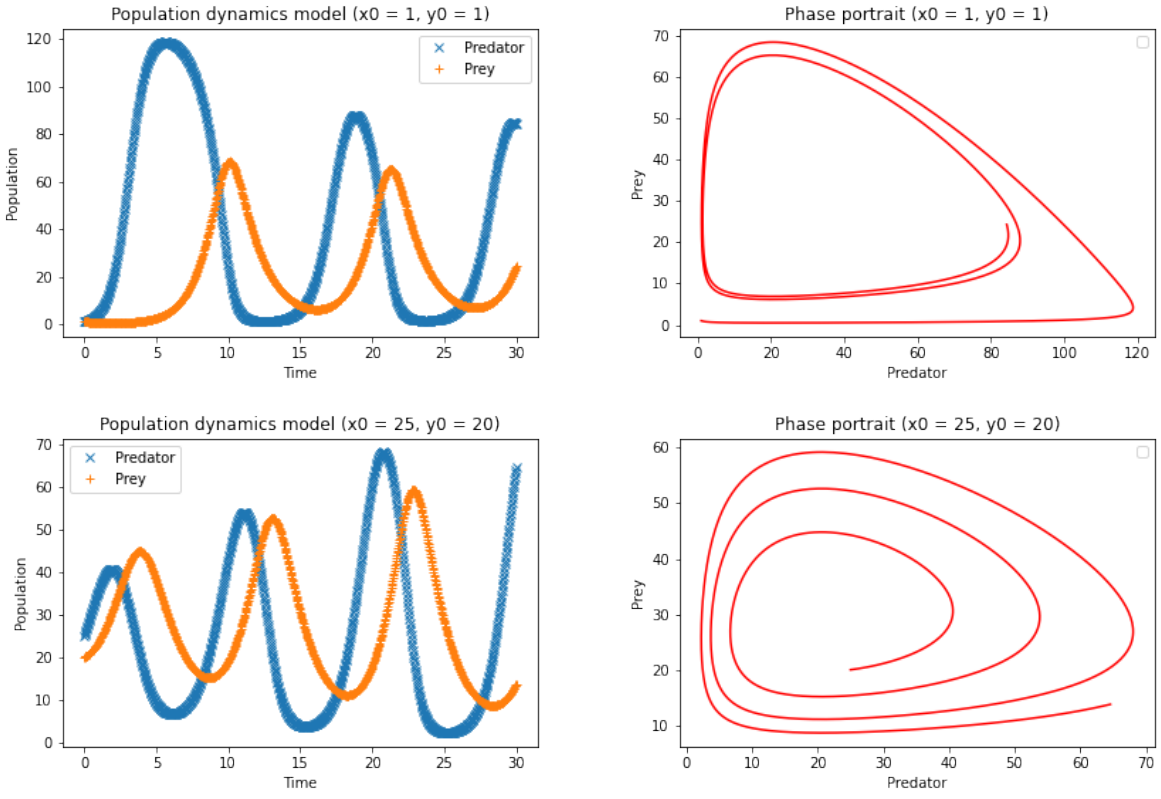
3.1 Population dynamics model

In the first part of the problem, we were asked to simulate a model of the following system of ODE's. In the context of our problem, the $\hat{x}$ resembles the predator and the $\hat{y}$ resembles the prey. We were then asked to find the equilibrium points of the following ODE equations:

$$\hat{x} = rx(1 - \frac{x}{k}) - \frac{axy}{c+x}, x \geq 0 \tag{1}$$

$$\hat{y} = \frac{baxy}{c+x} - dy, y \geq 0 \tag{2}$$

The equilibrium points I obtained from the following system were $(0,0), (20.588, 29.481) and (125,0)$. I was able to find these points through Desmos and analyzing the graphs at which the $\hat{x}$ and $\hat{y}$ were at 0. Lastly we were asked to plot the ODE dynamics at varying intial conditions in the time domain and this is what you will see pictured on the left. On the right we will see the phase potrait of the corresponding left time graphs which speaks to its dynamics. Consider the following graphs in Figure 9:



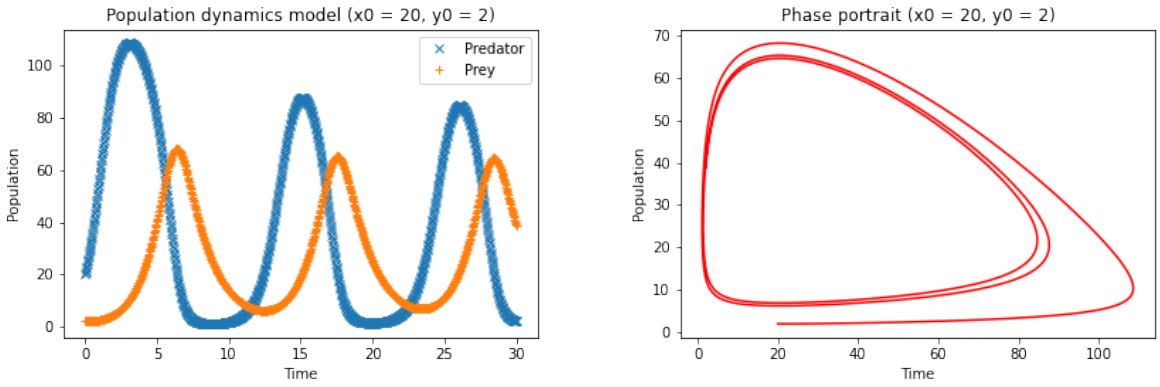


Figure 9: Predator Prey with at initial conditions

To conclude, we can acknowledge how the varying initial conditions causes the dynamics of these species to be very different. However this makes sense as different start points would have drastic changes in the population dynamics. The graph I found most interesting was the model with initial conditions ($x0 = 20, y0 = 2$). We can see that the lack of prey has caused the second oscillation to dip down before finding that point at which it constantly oscillates. similarly at initial conditions ($x0 = 25, y0 = 20$), we actually see both species blossom and see their populations grow over time. However having some previous knowledge about this model I know that this model eventually flattens out and oscillates at a constant value similar to the results described in the i.c. of ($x0 = 20, y0 = 2$).