Curve Fitting

After completing this section, students should be able to:

- Use technology to fit an exponential function to data and use it to extrapolate.

Introduction

Last time we looked at plots of early incidence data for covid and talked about fitting exponential curves to these plots.

Today, we'll learn to make these plots and fit curves in Python.

Please follow along on the Google Colab document at tinyurl.com/math115unc > Week 12 > exponentialGrowthUSDataPython.ipynb

You can make a copy of the file to work with using File ¿ Save a copy in Drive

Sketch of coding instructions for plotting

- Load the following tools, using this bit of code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.optimize as opt
```

- You will need up upload any data files.
  - First, locate the data file in tinyurl.com/math 115unc ¿ Week 12
  - Next, click on the file name to open up the file in Drive
  - Download the file using File ¿ Download ¿ Comma separated file (.csv, current sheet)
  - Locate the file on your computer, probably in your Downloads folder
  - Now, in colab, click on the folder icon on the far left and then the upload icon.
  - Locate the .csv file that you just downloaded and upload it.
  - Rename the file that you just uploaded, to something simple, if needed using the three dots to the right of its name.
- You can read in data with the following code (fill in the actual filename).

```
myData = pd.read_csv("FILL_IN_FILENAME.csv")
```

- You can plot it as follows.

```
plt.scatter(FILL_IN_VARIABLE_CONTAINING_XVALUES, FILL_IN_VARIABLE_CONTAINING_YVALUES)
```

```
plt.suptitle('UPPER_TITLE')
plt.title('LOWER_TITLE')
plt.xlabel('X_AXIS_LABEL')
plt.ylabel('Y_AXIS_LABEL')
```

- To log transform values, use:

  ```
  np.log(X_VALUES)
  ```

Curve Plotting in Groups

In groups, use Python plots to answer the following questions. Skeleton code is in tinyurl.com/math115unc > Week 12 > plotCovidDataForStudentsToComplete.ipynb

Question. Was North Carolina experiencing exponential growth of cases between March 20 and May 31?

Question. Which of these countries was experiencing the closest to exponential grown during the month of March (days 61 through 91, written in Python as [60:91])?

A. Israel

B. Italy

C. Sweden

D. USA

Sketch of coding instructions for fitting a line

Here is code to fit a line to data.

- Load in tools

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.optimize as opt
```

- Upload data file. Click on the folder icon on the far left and then the upload icon.
- You can read in data with the following code (fill in the actual filename).

```
myData = pd.read_csv("FILL_IN_FILENAME.csv")
```

- You should plot the log transformed data to check it is linear. You might need to select just some of the rows, as in the following.

```
xx = myData.Days[60:80]
yy = myData.Cases[60:80]
zz = np.log(yy)

plt.scatter(xx, zz)
```

- Define a linear function with parameters for slope m and y-intercept b

```
def linearFn(x, m, b):
  return m*x + b
```

- To fit linearFn to data and find the best values of the parameters m and b, we use the expression:

```
(m1, b1), pcov = opt.curve_fit(linearFn, xx, yy)
```

  where xx is the x-values and yy is the y-values of the data we want to fit.

  This outputs the best fit parameters into the variables $m1$ and $b1$. It also outputs the "covariance" into the variable pcov, which tells how good the fit is, but we will ignore this for now.

- Plug the x-values into the linear function to get the predicted values for log(cases) (here xx = myData.Days[60:80])

```
predictedzz = linearFn(xx, m1, b1)
```

  or you can do it this way:

```
predictedzz = m1*xx + b1
```

- Now take exp to get the predicted cases

  ```
  predictedyy = np.exp(predictedzz)
  ```

  or you can do it this way:

  ```
  predictedyy = np.exp(m1*xx + b1)
  ```

- Plot the x values and predicted values as a check. Here we plot the log transformed data and the line.

  ```
  plt.plot(xx, predictedzz)
  plt.scatter(xx, zz)
  ```

- Here we plot the original data and the exponential curve.

  ```
  plt.plot(xx, predictedyy)
  plt.scatter(xx, yy)
  ```

- Here is another way to do the same thing:

```
xx = myData.Days[60:80]

plt.plot(xx, m1*xx + b1)

plot(xx, np.exp(m1*xx + b1))
```

Tasks for Curve Fitting in Groups

- In groups, fit a line to log transformed data for Israel for the month of March. Skeleton code is at tinyurl.com/math115unc > Week 12 > fitLineForStudentsToComplete.ipynb

- Plot the log transformed data for Israel for the month of March and a fit line.

- Plot the original data and the corresponding exponential curve.

- Predict the number of covid cases in Israel on April 2 if this trend continues and compare to the true number of cases in Israel on April 2 (which is in the data file).