Tutorial 03 Time

This tutorial consists of two data explorations. The first uses data that is relatively similar to the data in the lecture notes and examples. The second is more unstructured and open ended.

We will be using **python notebooks** (.ipynb files) to do the explorations. You can run these in a number of ways including Jupyter, if you are familiar with that software. However, in the labs I recommend that you use Colab from Google. **For more details see Tutorials.pdf or watch the "How to get started with …" videos from Moodle.**

Naming: For each exploration you should create a notebook and save it when you have finished. You should name the two notebooks Tut03-A.ipynb and Tut03-B.ipynb.

Structure: Every numbered item in the exploration should have a code section and a markdown section underneath where you discuss your findings. There should also be a code section at the top of the Notebook with the imports.

Exploration A

The company who supplied the Products data in the lecture notes also want an investigation into their website. As last week, the data (https://tinyurl.com/ChrisCoDV/Pages/DailyHits.csv) shows the number of page hits per day.

As in the lecture examples the company wants to explore the page hit trends over the year. Last week you should have identified the high volume and medium volume pages, with the low volume being all the others. It was your choice how high, medium and low were defined, but for the purposes of this exploration pick the top 2 as high volume and the next 8 as medium volume (this is very likely to be the choice you made last week).

1. Create an initial exploratory chart showing line plots for all pages using one of the examples from the lecture. [**Hint**: there is one (obvious) item that won't fit in the plot – remove it by commenting out a line of code by putting a # symbol at the start of the line].

2. The high volume pages are not particularly interesting as most visitors to the site use these. Instead the company is interested in driving up engagement with the site by getting users to explore further. They want to focus on medium volume pages, so create a chart showing line plots for these with a legend indicating which is which. [**Hint**: there are 8 medium volume pages – you should have identified them last week.]

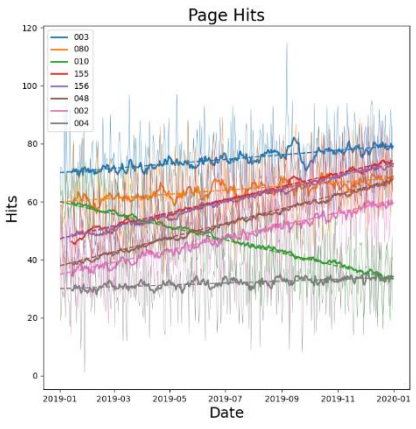
3. The previous chart is a little overcrowded, so create a further chart for medium volume pages which also includes a 14-day rolling average for each time series.

4. Now create a further chart for medium volume pages which also includes a trendline (so it should show the original time series, 14-day rolling average and trendline combined).

5. Finally make the plot a little easier to interpret as follows:
    a. Make the line width a little thinner for the original time series but not so thin that it can't be seen.
    b. Make the trendline a dashed line rather than continuous. [**Hint**: search the lecture notes for the linestyle parameter.]
    c. Manually reorder the selected pages so that the order of items in the legend (top to bottom) matches the visual order of the lines in the chart.

    [Hint: if you plot the rolling average lines *before* plotting the original time-series, matplotlib will show thick lines in the legend (as it picks the first set of lines to create the legend; also don't forget to reset the colours twice (before each new set of lines).]

    Your final chart should look similar to the one on the right.



Exploration B

The file world_population.csv (available at https://tinyurl.com/ChrisCoDV/world_population.csv) contains data about population densities from 1960 to 2016. Note this is **population density** (i.e. the number of people per square kilometer) and not **absolute population**. So some of the smallest countries have the highest densities.

1. The data in this exploration also involves more work to get it into shape so read in the data and wrangle it as follows (slightly different to last week):

   First, the country name is a bit inconvenient to use to select specific countries so instead we will use the country code. However, that means we need to use column 1 as the index column and not column 0.

   Next, each column contains the data from a particular year, whilst each row contains the data for a country. We would like it the other way around to match the previous examples, so transpose it. [**Hint:** you should have done this last week.]

   Next, drop the initial rows which contain descriptive data – the rows you need to drop are 'Country Name', 'Indicator Name' and 'Indicator Code'. [**Hint**: you should have done this last week except dropping Country Code rather than Country Name.]

   You should also drop the rows '1960' and '2016' (you can do this at the same time as the other rows) which contain no data. [Note: if you do not do this the numpy.polyfit() function will not work in Jupyter Lab, although it does seem to work in PyCharm.]

2. Create a chart showing line plots for population densities for all countries over the period. Use the legend to work out the country codes of the top five countries and then comment out the legend.

3. Now create a chart showing just the top five countries by population density. Make sure that the y-axis is zeroed to get a fair comparison.

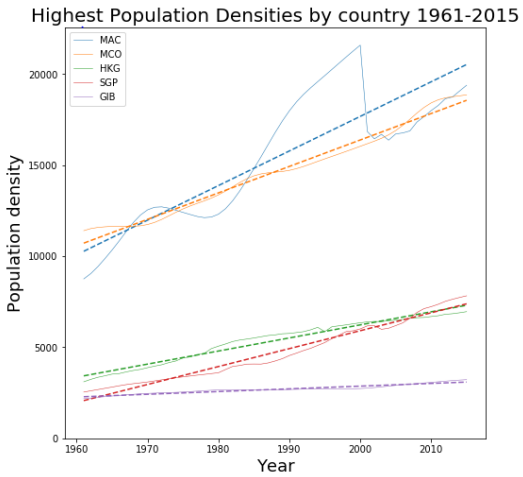
4. Next create the same chart but with a 10-year rolling averages included. Comment on whether you think that this chart helps the data exploration or not. [**Hint**: there is no right answer, you should just put your opinion.]

5. Next create the same chart but with trendlines (but not rolling averages). When you calculate the linear regression you will need the following line, rather than the one in the lecture notes and examples:

   z = np.polyfit(x, data[name].astype(np.float64), 1)

   Your final chart should look something like the one on the right.

   Comment on whether you think that this chart helps the data exploration or not. [**Hint**: again there is no right answer.]



6. Finally, something dramatic happened to the population density in one of these five countries in one particular year. What year was it and can you figure out (from Google) what happened to the country that year, even if that doesn't explain the change. [Note: you do not need a code section for this item.]