Code & Data

First you should download the VisCode.zip file from Moodle and unzip it – this contains all the code & data from the Visualisation lectures. You should use Google Colab, or install PyCharm Community Edition on your own computer downloaded from https://www.jetbrains.com/pycharm/ (if you are working in the University labs this should be already installed or available from the Software Centre). From PyCharm, use File > Open to open the VisCode folder as a project.

Visualisation 01 tutorial: Proportion & Time

This tutorial consists of two data explorations. The first uses data that is very similar to the data in the lecture notes and examples and gives you a chance to try out adaptations of the techniques used in the lecture. The second (optional) exploration involves more data wrangling and is more open ended.

Exploration A

The company who supplied the Products data in the lecture notes also want an investigation into their website. The data is in a similar format to the Products, showing the number of page hits per day, rather than the number of product sales, but there are many more web pages than products.

As with the lecture examples the company wants the pages classified into High Volume (used by most visitors to the site), Medium Volume (used by some visitors to the site) and Low Volume (only used occasionally). You will need to decide on what the classifications are. The company also wants to page hit trends over the year.

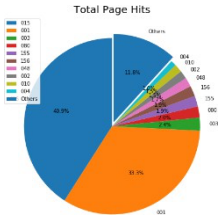
1. Read in the data from the file **Pages/DailyHits.csv** (instead of Products/DailySales.csv) and do some basis statistical analysis using the .head() and .describe() functions. Make a note of the details – e.g. what is the time period under investigation, how many pages are there, which page has the most hits, what is the smallest number of hits for any page, …

2. Create a bar chart showing the total number of hits for all pages. Most pages only have relatively few hits – how many pages have a significant number? Note: the chart will almost certainly be unreadable, so you will need to do this by sorting the data first and then printing out the first 15 entries or so using something like
```
data = data.reindex(data.sum().sort_values(ascending=False).index, axis=1)
print(data.sum().head(15))
```

3. Now by modifying the code in "03BarChart automatic.py", create a script that automatically classifies pages into high, medium and low volume pages. You will need to decide the boundaries for each type. Your code block should produce 3 bar charts.

4. Finally create a pie chart summarising your findings. The pie sections should be sorted in order of decreasing page hits (like the ones in the lecture are sorted in order of decreasing sales) and all the low volume pages should be grouped together into one pie segment.

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



5. Next create an initial exploratory time-series chart showing line plots for all pages.

6. 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.

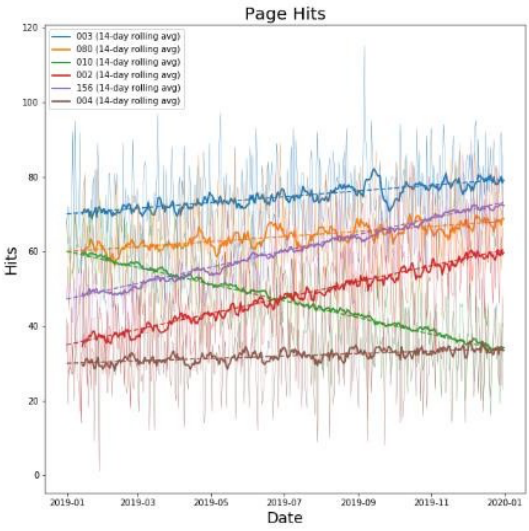
7. The previous chart is a little overcrowded, so create a further chart for medium volume pages which also includes a 14-day rolling average for each time series.

8. 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).

9. 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**: use 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.



Code & Data

You just need to download VisCode.zip from last week's links on Moodle (see Vis01 Proportion & Time - tutorial.pdf).

Visualisation 02 tutorial: Correlation & Dimension

This tutorial consists of a single data exploration. It uses data that is very similar to the data in the lecture notes and examples and gives you a chance to try out adaptations of the lecture techniques.

The company who supplied the Products data in the lecture notes also want an investigation into their website. The data file, **Pages/DailyHits.csv**, is in a similar format to the Products, showing the number of page hits per day, rather than the number of product sales, but there are many more web pages than products.

In addition, the company has provided a number of other data files in the same folder (containing page metrics that could be obtained from Google Analytics). All of these files show data averaged or totalled over the year:

- PageExitRate.csv shows, as a percentage, the average exit rate. For each page, this is the percentage of visitors who leave the site from that page (e.g. by clicking an external link or closing the browser tab).
- PageRevenue.csv shows the total revenue raised by serving adverts from each page.
- PageSize.csv shows, in Kilobytes, the size of the html content for each page.
- PageSpeed.csv shows the average time, in seconds, it takes for users to download each page.
- PageViewingTime.csv shows the average time, in seconds, that users spend looking at each page.

The company wants to investigate correlations between page hits over the year and also explore relationships between these metrics and the total hits for each page.

1. Create a scatter plot of the two high volume pages against each other (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 originally). Can you detect any correlation visually?

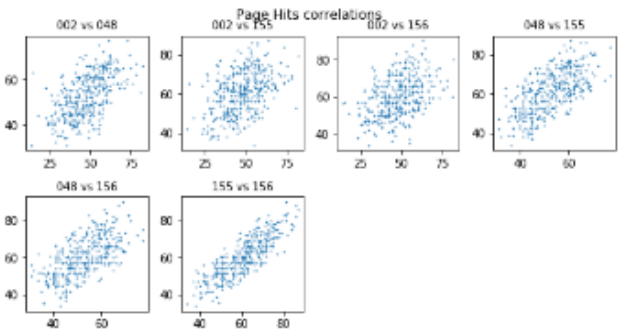
2. Create a single visualisation showing scatter subplots for all pairs of medium volume pages against each other. Which two pages are most strongly correlated with each other? Which other pages seem to have some correlation, visually?

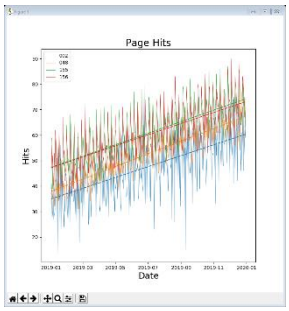
3. Now confirm your findings by creating a heatmap showing the correlations between all the high and medium volume pages. Comment on the strongest positive and inverse correlations.

4. After seeing the heatmap, the company has decided it is interested in all positive correlations where the Pearson coefficient is greater than 0.5. Identify the corresponding pages and create a visualisation showing scatter subplots for all such pairs of pages.

5. Now draw line plots with trendlines for the positively correlated pages you selected above.

   Your visualisations for items 4 and 5 should look something like those below.





6. Next read all the metrics data (from the other spreadsheets) in and compile a summary data frame, including the annual sum of page hits, similar to the lecture examples. Use this to create a heatmap of correlations between the metrics. What are the strongest correlations?

7. Create a visualisation showing radar subplots for high volume pages. You should decide what order the 6 metrics appear around the each plot (using the guidelines in the lecture and noting that the company regards viewing time as a neutral indicator – it is good that users spend a lot of time on each page but it may mean that they can't find what they are looking for). Comment on the order you have chosen.

8. Now create a visualisation showing radar subplots for medium volume pages. What you will notice is that some of the metrics are too small to distinguish from each other. This is because some of the metrics are being normalised by very large values from the high volume pages.

   Instead it makes more sense to normalise by the maximum value from medium volume pages only since the visualisation is restricted to these pages. To do this create a normalised data frame using:

   normalised_ data = summary_data / summary_data.loc[selected].max()

   where "selected" contains the list of medium volume pages.

   Comment on which page seems most valuable in terms of hits and revenue.

9. Finally the company is interested in how the viewing time varies across medium volume pages as compared with page hits and revenue. Generate a bubble plot of hits against revenue with viewing time determining the bubble sizes.

   In order to do this you will need to restrict the summary data to just medium volume pages. The easy way to do this is:

   summary_data = summary_data.loc[selected]

   where "selected" contains the list of medium volume pages.

   Given that you might expect that the page with the highest revenue might have the longest viewing time and / or the most hits, what feature (in terms of the combination of page hits, revenue and viewing time) stands out for this plot?

   Two of your plots should like something like the ones below.