Tutorial 06 Dimension

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.

Notebooks: We will be using Colab to create Python Notebooks on your Google Drive although you may also use Jupyter. Read the document Tutorials.pdf from Moodle to see how to set these up.

Naming: For each exploration you should create a notebook and save it when you have finished. You should name the two notebooks Tut06-A.ipynb and Tut06-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 well as the data explored over the last few weeks (https://tinyurl.com/ChrisCoDV/Pages/DailyHits.csv), there are a also a number of other data files in the same folder. All of these files show data averaged or totalled over the year under investigation:

- https://tinyurl.com/ChrisCoDV/Pages/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).
- https://tinyurl.com/ChrisCoDV/Pages/PageRevenue.csv shows the total revenue raised by serving adverts from each page.
- https://tinyurl.com/ChrisCoDV/Pages/PageSize.csv shows, in Kilobytes, the size of the html content for each page.
- https://tinyurl.com/ChrisCoDV/Pages/PageSpeed.csv shows the average time, in seconds, it takes for users to download each page.
- https://tinyurl.com/ChrisCoDV/Pages/PageViewingTime.csv shows the average time, in seconds, that users spend looking at each page.

These page metrics are typical of website data and, for example, can be obtained for a site that you own using Google Analytics.

As in the lecture examples, the company wants understand relationships between these metrics and between the total hits for each page (which can be obtained by summing the values in DailyHits.csv). In tutorial 02 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 originally).

1. First read all the data in and compile a summary data frame, similar to the lecture examples.

2. Create an initial visualisation showing bar chart subplots for all 6 metrics (i.e. those from the 5 files plus the total hits). Comment briefly on any likely correlations.

3. 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.

4. 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 (assuming you have adapted one of the lecture examples) 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.

5. Next create a correlogram / pair-plot of all 6 metrics. Comment on the obvious correlations.

6. Unfortunately the correlogram is a bit unclear for some subplots because again the total hits for the high volume pages are so much larger than the other pages. To explore the correlations further, create a heatmap of correlations between the 6 metrics, similar to those generated for lecture 04. Comment on the correlations that you identify.

7. 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.

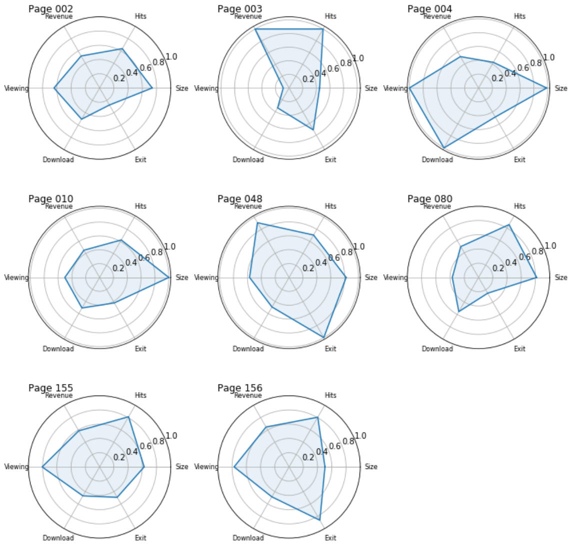
   In order to do this you will need to restrict the summary data to just medium volume pages. The easy way to do this is:

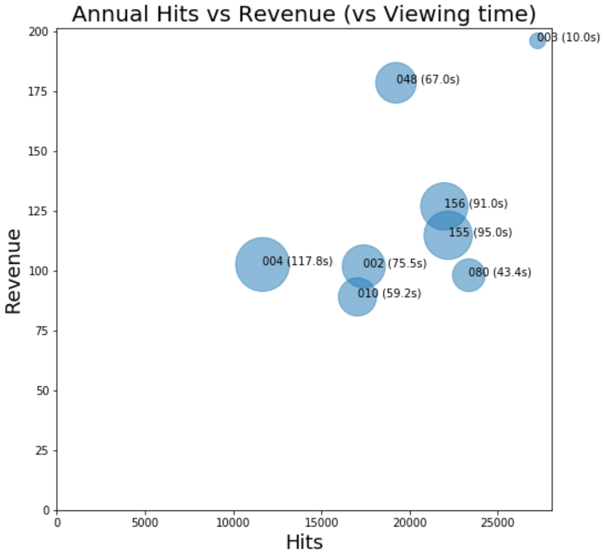
   summary_data = summary_data.loc[selected]

   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.





Exploration B

Once again the second exploration looks at the Iris dataset (see Tutorial 04 for more details).

This week you will look comparing the data across the 4 different dimensions (i.e. the measurements).

1. As last week, the data in this dataset has no obvious index column (recall that an index column usually has a unique identifier in each row). So first read the data in and check it is in the right format – there should be 5 columns (4 for the measurements and 1 for the variety) and 150 rows.

2. Next create a summary dataframe. To do this first create an empty dataframe

   ```
   summary_data = pd.DataFrame()
   ```

   Next add 3 columns, one for each variety, containing the mean measurements for that variety. Here is the code for Setosa variety:

   ```
   summary_data['Setosa'] = data[data['variety'] == 'Setosa'].mean()
   ```

   This will create a summary dataframe with 3 columns (one for each variety) and 4 rows (one for each metric). However by analogy with the lecture and the previous exploration it makes more sense for each metric to have its own column and each row to represent a variety. To achieve this just transpose the dataframe:

   ```
   summary_data = summary_data.transpose()
   ```

   Check it's what you want by printing it out.

3. Next create a visualisation with 3 radar subplots, one for each variety, and with each radar plot showing the mean lengths & widths for that variety.

   You can virtually use the code from the lecture example 05 unchanged for this, however:

   - The selected list needs to be a list of the varieties, ['Setosa', 'Versicolor', 'Virginica']
   - The lengths & widths are all more or less the same size. If they are normalised (as in the examples) it actually makes the visualisation harder to understand. Therefore just visualise the unnormalised data, either by modifying the visualisation code or, even easier, just by setting the normalised dataframe to be the same as the unnormalised one, i.e. normalised_data = summary_data
   - If you are not using normalised data, you will need to change the y-axis limit and tick marks to something sensible matching the range of values.

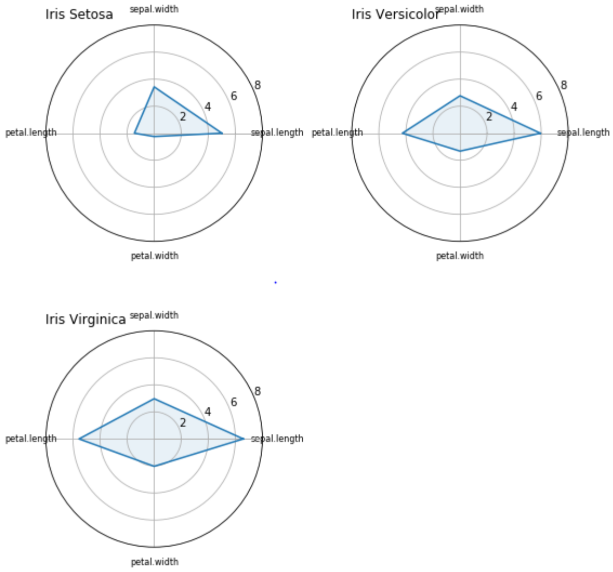
   Comment on the radar subplots and what they show.

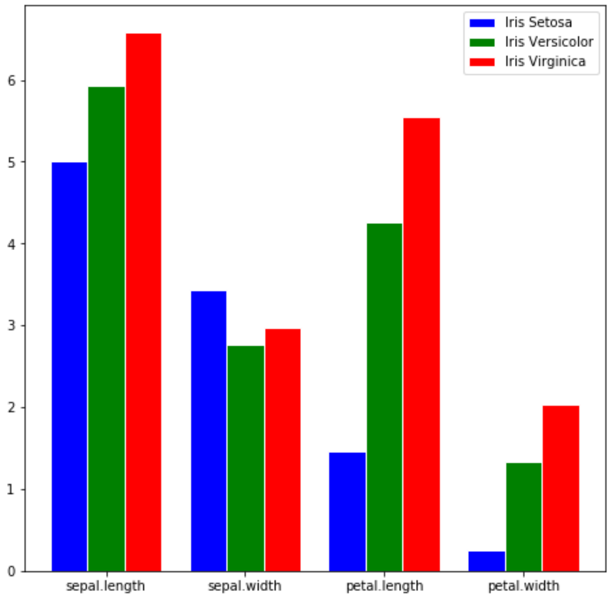
4. In fact it might be easier to use a comparative bar chart to compare these metrics.

   You can use the code from the lecture example almost unchanged. Once again, don't normalise the data and adapt the y-axis tick marks and legend labels).

   Comment on whether you think the radar subplots or the comparative bar chart is easier to interpret. [**Note**: there is no right answer.]

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


Iris Setosa
sepal.width
sepal.length
petal.length
petal.width
Iris Versicolor
sepal.width
sepal.length
petal.length
petal.width
Iris Virginica
sepal.width
sepal.length
petal.length
petal.width


Iris Setosa
Iris Versicolor
Iris Virginica
sepal.length
sepal.width
petal.length
petal.width