K-Pop Project - Documentation

**Introduction**

I am a BBA + QSS double major, and I am interested in the Asian entertainment industry. For my final project, I combined these two interests and decided to look into K-Pop [datasets](https://www.kaggle.com/datasets/kimjihoo/kpopdb) that I discovered on Kaggle. I found initial explorations ([1](https://www.kaggle.com/code/mpwolke/kpop-idols-boy-groups)) ([2](https://www.kaggle.com/code/kerneler/starter-k-pop-database-1992-2020-918cbe92-5/notebook)) on this data, but they were not comprehensive at all in terms of looking at correlations between relevant variables in the data. I looked into the data further, and this documentation details the steps I took to reach a few findings.

I took a deeper dive into Python pandas in order to examine the data. I had never used Python before this class, so this required a lot of research before I could draw any findings. I have only worked with R more extensively, so I know some of the possibilities when coding with data.

**Process Steps**

Based on the existing exploration in the datasets, I installed plotly to create pretty visualisations.

1. **Cleaning Data**

An important part of data is cleaning it for analysis. I set out to do exactly that. First, explored the data using summaries to see which variables were there, and what kind of data types the variables were.

df\_idols = pd.read\_csv('kpop\_idols.csv')

df\_boyg = pd.read\_csv('kpop\_idols\_boy\_groups.csv')

df\_girlg = pd.read\_csv('kpop\_idols\_girl\_groups.csv')

df\_vids = pd.read\_csv('kpop\_music\_videos.csv')

df\_idols.head(5)

df\_idols.info()

I recognised that the df\_boyg and df\_girlg both had data on which group the individuals belonged to, but the variables were labelled differently. I changed the name of the variable to match, then [merged the datasets](https://jakevdp.github.io/PythonDataScienceHandbook/03.07-merge-and-join.html) so that I could have more data to work with per dataset.

I created two main datasets that I would then use; one for boy groups, and one for girl groups.

df\_boyg.rename(columns = {'Name':'Group'}, inplace = True)

df\_girlg.rename(columns = {'Name':'Group'}, inplace = True)

df\_bgs = pd.merge(df\_boyg, df\_idols, on='Group')

df\_ggs = pd.merge(df\_idols, df\_girlg, on='Group')

There were only two numerical variables, so I converted the dates to date variables. I also added a column for debut age based on calculation, as this variable will come in useful for examining data.

df\_bgs["Debut"] = pd.to\_datetime(df\_bgs["Debut"])

df\_bgs['Date of Birth'] = pd.to\_datetime(df\_bgs['Date of Birth']) #changing data type of DoB to date / time

df\_bgs = df\_bgs.assign(Debut\_Age = (df\_bgs['Debut'] - df\_bgs['Date of Birth']).dt.days) #adding extra col for debut age

A screenshot of a computer

Description automatically generated

1. **More Data**

I was interested in finding out more based on segmenting the data. I segmented the data between:

* 1. Male/female idols whose country of origin is South Korea, versus not. I wanted to explore the differences between domestic and international idols.
  2. Male/female idols in the Big 3 versus not. In the K-Pop industry, there are three large companies that dominate most of the market share. I wanted to compare these companies with smaller ones.

df\_bgs\_korea = df\_bgs[(df\_bgs['Country'] == "South Korea")]

df\_bgs\_intl = df\_bgs[(df\_bgs['Country'] != "South Korea")]

df\_bgs\_big3 = df\_bgs[(df\_bgs['Company'] == "SM") | (df\_bgs['Company'] == "JYP") | (df\_bgs['Company'] == "YG")]

df\_bgs\_other = df\_bgs[(df\_bgs['Company'] != "SM") | (df\_bgs['Company'] != "JYP") | (df\_bgs['Company'] != "YG")]

1. **Creating Charts**

First, I used the same exploration technique another data scientist attempted in their analysis. The difference was that mine had extra ‘numerical’ variables with the dates, so there are more charts from the same code!

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Prior to this project, I had no idea how to draw [trendlines](https://widu.tumblr.com/post/43624347354/matplotlib-trendline). I wanted to draw trendlines for the the scatterplots I had, so this required some troubleshooting. I encountered a few issues as below by running regression with dates. Finally, I was successful!

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Apart from simply scatterplots, I wanted to create more meaningful charts. Looking back to the data, I realised I could create histograms to look at the data generally, in a different way than the line charts above, and also from pie charts based on number (count) of groups in each company.

Graphical user interface, website

Description automatically generated

As you can see, I had quite a bit of fun exploring colours for the charts! For the pie chart below, I also wanted to change the colour of the title and labels, so that they can be readable on any program. The differences between the next three exhibits shows my progress during the project before I finally reached an end product I was happy with. I also played around with the legend positioning and background colour and opacity.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

Another visualisation can be on Jupyter Notebook on browser. I loaded my project there to see the result.

Graphical user interface, text, application, email

Description automatically generated

**Reflection and Next Steps**

Now that this project is done, I want to reflect on what I have learnt and consider some Python-related changes I could explore.

What I learnt

* Cleaning data
  + Converting dates to date
  + Merging datasets
* Changing type of data on graph (scatterplot, histogram etc)
* Decorating graphs (title, axes, colours etc)
* Adding trendlines to graphs

Next steps

* Add back to Kaggle, upload to Github
* Could add an API key and grab up to date information that way instead of using dataset, or an API that retrieves data, which I would convert to a csv. Using the same names but changing the dataset used, my code could easily be reused.
* Look at other available datasets and test more predictions
* Convert dates from days to age

**Works Cited**

Datartist. “K-Pop Database (1992-2020).” *Kaggle*, 3 June 2020, https://www.kaggle.com/datasets/kimjihoo/kpopdb.

Kerneler. “Starter: K-Pop Database (1992-2020) 918CBE92-5.” *Kaggle*, Kaggle, 3 June 2020, https://www.kaggle.com/code/kerneler/starter-k-pop-database-1992-2020-918cbe92-5/notebook.

Mpwolke. “KPOP Idols - Boy Groups.” *Kaggle*, Kaggle, 6 June 2020, https://www.kaggle.com/code/mpwolke/kpop-idols-boy-groups.