**Dennis M.**

08/08/2018

Hi guys, I am doing the TMDb project and am trying for hours to 'explode" the 'genres' column. I want to have one new line for every mentioned genre - all other information duplicated. I read every post with regards but none of the solutions worked for me. How did you do it? A hint is much appreciated... Many thanks!!

* **George L.**

**Mentor**

1:07 AM

there are two ways to do this. first, you can transform the data from wide to long after spliting the composite field "genres" into multiple fields, refer to this article for more info: [**http://www.datasciencemadesimple.com/reshape-wide-long-pandas-python-melt-function/**](http://www.datasciencemadesimple.com/reshape-wide-long-pandas-python-melt-function/); second, you can split the field, stack it and join it back, more info here: [**https://stackoverflow.com/questions/33622470/fast-way-to-split-column-into-multiple-rows-in-pandas**](https://stackoverflow.com/questions/33622470/fast-way-to-split-column-into-multiple-rows-in-pandas)

* Friday, Aug 10th

**Dennis M.**

9:34 PM

Thanks a lot, George. I had seen these posts. Tried the 'slow' aspproach of the poster today and it worked, any 'fast' approach however did not. Anyway, I did it. Thank you for your help!!!

* 4:01 AM

well, I am stuck again... with 'df\_new.groupby(['release\_year', 'genres\_new']).mean().popularity.groupby('release\_year').idxmax()' I compute the most popular genres per year and with '.max()' at the end I compute the corresponding value... BUT somehow I cannot combine genres and popularity value, which I would need to plot it appropriately. Any idea? Many thanks!!

**George L.**

**Mentor**

7:58 AM

what happened exactly when you say you cannot combine genres and pop value?

* Sunday, Aug 12th

**Dennis M.**

10:09 PM

df\_new.groupby(['release\_year', 'genres\_new']).mean().popularity.groupby('release\_year').idxmax() gives me a table with the following information:

release\_year 1960 (1960, Thriller) 1961 (1961, Animation)

df\_new.groupby(['release\_year', 'genres\_new']).mean().popularity.groupby('release\_year').max() gives me a table like that:

release\_year 1960 0.811910 1961 2.631987

How can I combine these? I just do not get it. Many thanks!

**George L.**

**Mentor**

7:53 AM

instead of combining the two dataframes, you can do this: df.groupby(['release\_year', 'genre\_new'])['popularity'].mean().reset\_index().groupby('release\_year').apply(lambda grp: grp.nlargest(1,'popularity'))

* sorry, i meant two series

**Dennis M.**

8:37 PM

Awesome! Thanks a lot!!

new

* **Abdel-Rahman E.**

5:54 PM

hello i need to ask you something

* 6:12 PM

release\_year 1960 Thriller|Music|Comedy|Crime

* I'm trying to get function to get maximum repeated genres in a year. this is the best what i got is this correct? unique\_genres = clean\_df.groupby(['release\_year'])['genres'].max()
* **George L.**
* **Mentor**
* 7:12 PM
* you wouldn't be able to use .max() since the genres column is a categorical one, should be .count()...

NEW IMPORTANT

**Revanth Kumar G.**

08/13/2018

Question on tmdb-movies.csv dataset: 'cast’ and ‘genres’, contain multiple values separated by pipe (|) characters.

* **Revanth Kumar G.**

10:47 AM

How to deal with it?

**Abdel-Rahman E.**

2:54 PM

I had the same question but i continued with the project and didn't split them and worked fine with me. If this is your question "should i split them or not?"

**George L.**

**Mentor**

5:51 PM

you can use .str.split() method: [**https://stackoverflow.com/questions/14745022/how-to-split-a-column-into-two-columns**](https://stackoverflow.com/questions/14745022/how-to-split-a-column-into-two-columns)

NEW

**Xia C.**

08/14/2018

hello. I am Shirley from Toronto. I am doing the TMDb Project. I divided voting into different levels, and want to find out the top genres for each level. but my plot looks like this. Can someone tell me how to filter only the top 3 genres for each voting level

* **Bogdan R.**

2:28 AM

[**@XiaC**](https://classroom.udacity.com/nanodegrees/nd002/parts/16904efb-1891-4822-ac1b-e6d74d746872/modules/bce8d241-89f4-422e-abda-c5a4e5ceb8fe/lessons/3176718735239847/concepts/54201485780923) , it doesn't look like you can post images or code on this platform, so it might be hard for people to answer your question. perhaps you should try slack as well, you might have better luck

**Xia C.**

thanks

* Yes, I was trying to past the graph, but there is no such option her

**George L.**

**Mentor**

you can also post a link to your GitHub repo

NEW

**Xia C.**

08/14/2018

hello. I am Shirley from Toronto. I am doing the TMDb Project. I want to see the genre trend by generations, I got stuck at grouping years by decade. anyone who know how to do that? thanks

* **George L.**

**Mentor**

6:25 PM

you can use pd.cut: [**https://stackoverflow.com/questions/45751390/pandas-how-to-use-pd-cut**](https://stackoverflow.com/questions/45751390/pandas-how-to-use-pd-cut)

* Thursday, Aug 16th
* **Scott M.**

12:08 AM

Xia, you could go back to the red-wine, white-wine code examples we worked on previously; add a new column for decade; then add in those decades based on release year.

NEW

**Brian T.**

Last Thursday

Okay, I'm totally stumped on a groupby problem for my project. I have a column of genre that I want to group by.

Then, I have another column where I've created bins and each row has a bin value of Low, Medium, or High.

Last, I have a column of profitability.

I want a dataframe that has the genre as the index, then a column of mean profitability, and finally, three more columns, one for each bin name, with the count of the number of instances that genre is rated at each bin level.

I can do a groupby df with the means of profitability, but I can't figure out how to do one with a column counting each of the bin values for that genre. And once it's done, can I then merge it with the df that has the means?

* **Brian T.**

1:57 AM

I think the bin groupings starts something like this: '''df\_genresplit\_pop = df\_genresplit.groupby(['genres'])['popularity\_levels'].'''

* I apologize, my code identifiers aren't working...
* df\_genresplit\_pop = df\_genresplit.groupby(['genres'])['popularity\_levels'].
* Update: I have noticed that value\_counts give me the result within the same column but I want each of the three values (Low, Medium, High) to each get their own column next to that genre, with the count of each value in its respective column.
* I discovered .unstack(level=1), which I think does the trick, I just need to merge it with the table that has the genre profitability means

**George L.**

**Mentor**

to count the members in a group, you can use .count() after groupby

* right, in your case, it should be value\_counts since you're grouping by genres and counting the cases within genre groups

NEW

**Brian T.**

Last Thursday

I've run into a brick wall with multiple-variable exploration. The feedback says: "This can be like correlation of popularity and budget together on revenue, you can compare their correlations also." I don't understand what this means or how to do it. I have two axes - one is the values and the other is the labels. I can't figure out how to make this a three-dimensional analysis that factors in popularity, budget, and revenue. If I can at least get guidance on how I should compare something like that, I can try and figure out the coding and visualization... Nothing I'm reading translates to this problem well enough for me to figure it out on my own.

* **George L.**

**Mentor**

5:25 AM

it's not clear what the reviewer means to me either...but to me, multiple variable analysis can just be a scatter plot between two numeric variables, or a bar plot between a numeric and a categorical variable

**Brian T.**

10:56 PM

I did that with a scatter plot and I got an approval so I guess it's good! Thanks for the advice.

**George L.**

**Mentor**

11:44 PM

good work!

NEW

**Yusuf B.**

Last Thursday

Hello, I have read this thread for insight, but I am still very lost with showing the most popular genre by year. so far, I am able to use nlargest to show the most popular title along with the index, but it doesn't break it out by year. instead, it just gets the most popular from all time:

* **George L.**

**Mentor**

5:18 AM

try this instead: df.groupby('release\_year').apply(lambda grp: grp.nlargest(1,'popularity'))

* so basically, we group by year, then for each group, we find the largest entry based on popularity

NEW

**Kamakshi P.**

Last Thursday

Hello, I received some feedback from my reviewer. He mentioned: 1. 1d or single variate exploration is still missing in the project, and 2. Limitations are still missing. I thought I took care of these points in my project, but I guess I didn't. Would anyone be able to give me an example of each of these so that I can have a better

**George L.**

**Mentor**

12:25 AM

for single variable exploration, you can use a histogram or boxplot to explore the distribution of a variable; for limitations of analysis, consider: whether the sample size is large enough, whether the conclusion is generalizable to the larger population, and what the effect of removing null values/duplicates is etc.

NEW

**Abhay Rajesh S.**

Today

Hi, I got my project review and I got that i need to add a function to minimise the repetitiveness but there is nothing which is getting repeat and i don't understand where and what function to add . Any help would be appreciated

* **George L.**

**Mentor**

7:44 AM

in that case, you can find a small chunk of code and define a function.

* refer to these links:
* [**https://stackoverflow.com/questions/3898572/what-is-the-standard-python-docstring-format**](https://stackoverflow.com/questions/3898572/what-is-the-standard-python-docstring-format)
* sorry, that's the only one :)

new

**Xia C.**

7:23 AM

def cut\_level(): """devide budget, revenue, runtime into different levels """ columns=['budget','revenue', 'runtime','vote\_average'] for column in columns: min\_value=df[column].min() first\_quantile=df[column].quantile(0.25) second\_quantile=df[column].quantile(0.5) third\_quantile=df[column].quantile(0.75) max\_value=df[column].max() bin\_edges=[min\_value, first\_quantile, second\_quantile, third\_quantile, max\_value] bin\_names=['low','medium','mod\_high','high'] return df[column +"\_level"]=pd.cut(df[column], bin\_edges, labels=bin\_names)

cut\_level()

some one who can help me check this code? I want to define a function to cut levels to avoid using duplicated coding. it says SyntaxError: invalid syntax

NEW

6

**Mira S.**

6:44 PM

i have got my feed back for the project and i was told that i only used 2d plots, and need to have 1d plots, can somebody elaborate more?

* **Mira S.**

7:07 PM

is it just not to use matplotlib?

**George L.**

**Mentor**

no. they are referring to 1-dimension plot such as a histogram or boxplot.