**Extracting Nested Column Data using .stack()**

Let’s turn our attention to the Genres column. This is quite similar to the categories column but more granular.

Challenge

How many different types of genres are there? Can an app belong to more than one genre? Check what happens when you use .value\_counts() on a column with nested values? See if you can work around this problem by using the .split() function and the DataFrame's [.stack() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.stack.html).

.

.

..

…

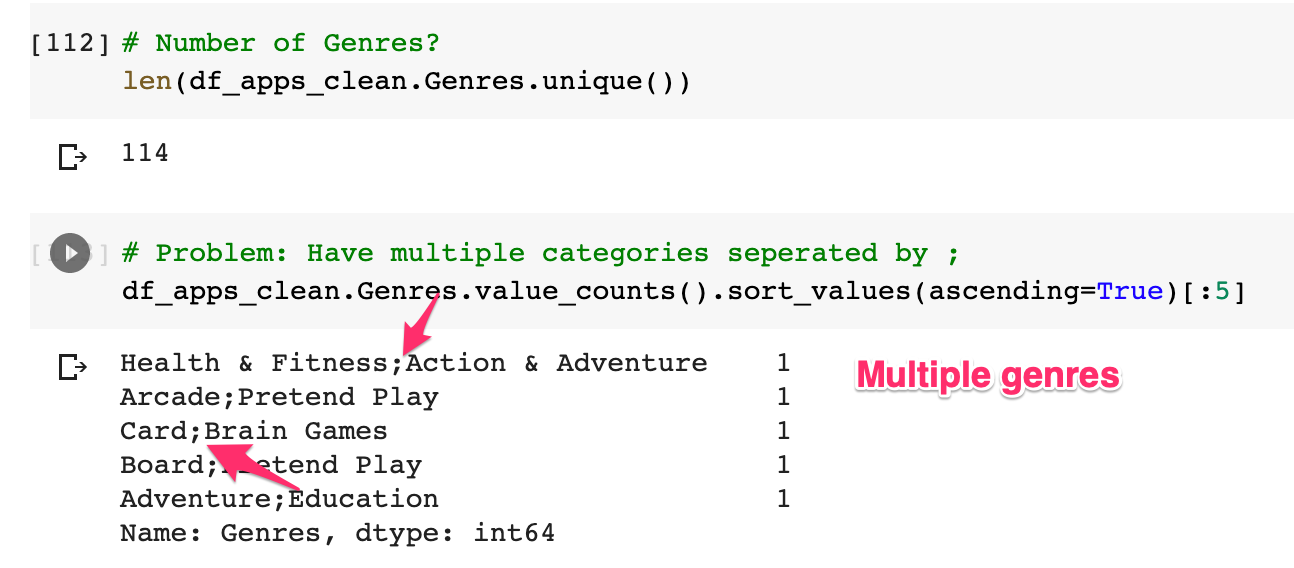
..

.

.

**Solution: Working with Nested Column Data**

If we look at the number of unique values in the Genres column we get 114. But this is not accurate if we have nested data like we do here. We can see this using .value\_counts() and looking at the values that just have a single entry. There we see that the semi-colon (;) separates the genre names.



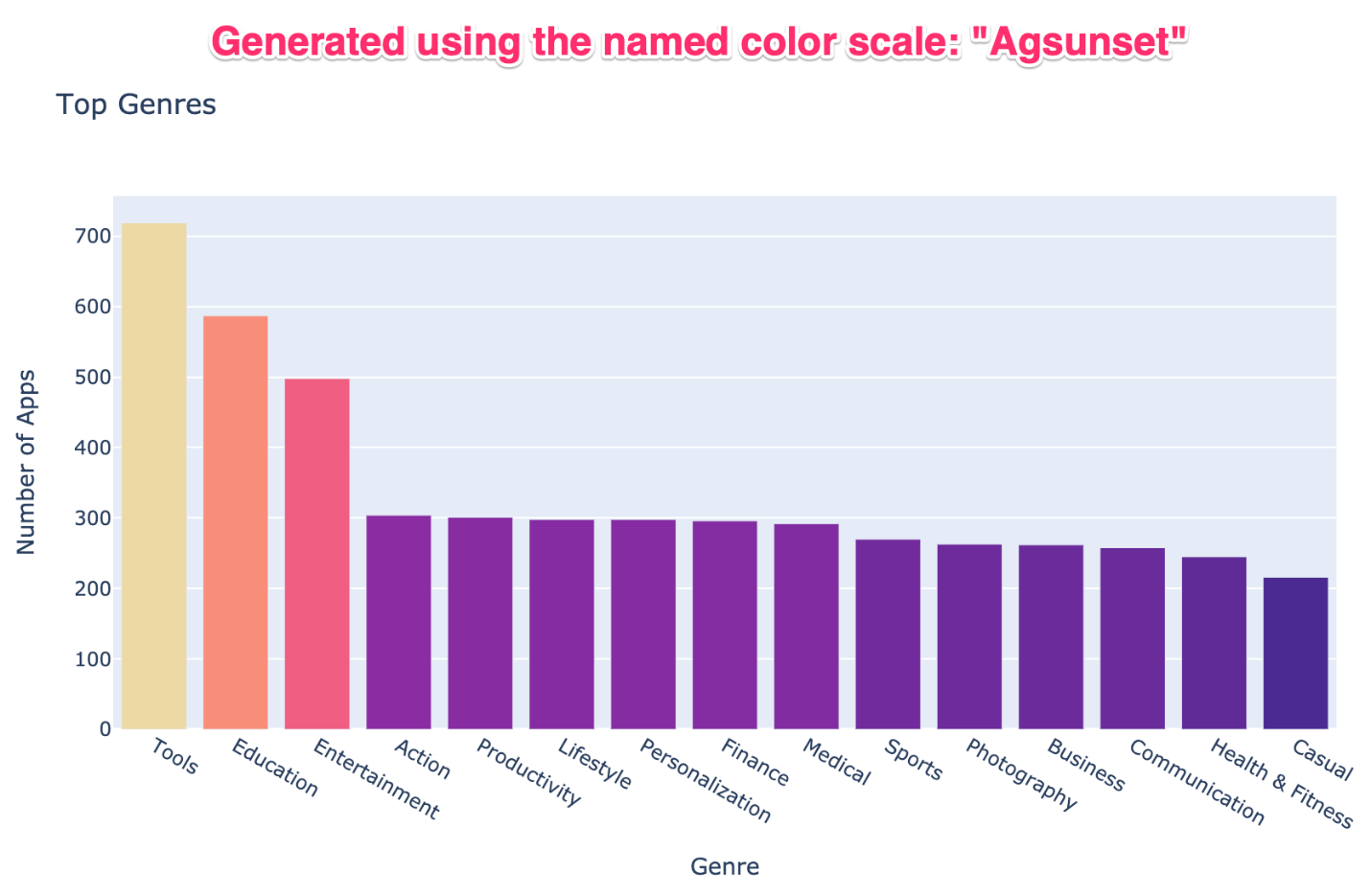
We somehow need to separate the genre names to get a clear picture. This is where the string’s .split() method comes in handy. After we’ve separated our genre names based on the semi-colon, we can add them all into a single column with .stack() and then use .value\_counts().

1. # Split the strings on the semi-colon and then .stack them.
2. stack = df\_apps\_clean.Genres.str.split(';', expand=True).stack()
3. print(f'We now have a single column with shape: {stack.shape}')
4. num\_genres = stack.value\_counts()
5. print(f'Number of genres: {len(num\_genres)}')

This shows us we actually have 53 different genres.

Challenge

Can you create this chart with the Series containing the genre data?



Try experimenting with the built-in colour scales in Plotly. You can find a full list [here](https://plotly.com/python/builtin-colorscales/).

* Find a way to set the colour scale using the color\_continuous\_scale parameter.
* Find a way to make the colour axis disappear by using coloraxis\_showscale.

.

.

..

…

..

.

.

**Solution: Working with Colour Scales in Plotly**

1. bar = px.bar(x = num\_genres.index[:15], # index = category name
2. y = num\_genres.values[:15], # count
3. title='Top Genres',
4. hover\_name=num\_genres.index[:15],
5. color=num\_genres.values[:15],
6. color\_continuous\_scale='Agsunset')
8. bar.update\_layout(xaxis\_title='Genre',
9. yaxis\_title='Number of Apps',
10. coloraxis\_showscale=False)
12. bar.show()