long-vs-wide-data

September 15, 2022

0.1 long-vs-wide-data

A dataset can be written in two different formats: wide and long.

A wide format contains values that `do not repeat` in the first column.

A long format contains values that `do repeat` in the first column.

This is a long format:

Product	Attribute	Value
A	Height	10
A	Width	5
A	Weight	2
B	Height	20
B	Width	10

The same data is a wide format would be:

Product	Height	Width	Weight
A	10	5	2
B	20	10	NA

[1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:

```python
crop = pd.read_excel('Somalia crop production_FSNAU.xlsx')
```

[3]:

```python
crop.head()
```

[3]:

```
   Country     Zone          Region District  Year     Crop Livelihood_System  \
0  Somalia  Central  Shabelle Dhexe   Jowhar  2021   Sesame     Agro Pastoral
1  Somalia  Central  Shabelle Dhexe   Balcad  2021  Sorghum     Agro Pastoral
2  Somalia    South  Shabelle Hoose  Afgooye  2021  Sorghum     Agro Pastoral
3  Somalia    South  Shabelle Hoose  Afgooye  2021   Sesame     Agro Pastoral
4  Somalia    South  Shabelle Hoose  Baraawe  2021    Maize     Agro Pastoral
```

```
     Season  Production
0  2. Deyr         1.0
1  2. Deyr        20.0
2  2. Deyr      1600.0
3  2. Deyr       200.0
4  2. Deyr       120.0
```

Creating Pivot tables with Pandas.

index = column to groupby on the row axis

columns = column to groupby on the columns axis

values = column to aggregate

aggfunc = type of aggreagation function to use by defualt np.mean is used, for summation use np.sum

margins = Adds subtotal/grandtotal rows and columns: takes True or False values.

margins_names = Name of the row / column that will contain the totals when margins is True

pd.options.display.float_format = '{:.2f}'.format

[4]:

```
pivot = crop.pivot_table(index = "Year", columns = "Season", values =␣
 ↪"Production",aggfunc = np.sum, margins = True, margins_name = 'Grand Total')
```

[5]:

```
pivot.round(2)
```

[5]:

```
Season          1. Gu    2. Deyr  Grand Total
Year
1995        151284.63  104505.73    255790.35
1996        202766.07   46812.20    249578.27
1997        217353.09   60532.05    277885.14
1998        106053.38   92150.71    198204.09
1999        128883.00  130001.00    258884.00
2000        212307.00  109927.00    322234.00
2001        115394.50  148276.00    263670.50
2002        208929.00  181182.00    390111.00
2003        156305.90  129188.00    285493.90
2004        125309.00  162829.20    288138.20
2005         78548.00   68093.50    146641.50
2006        116257.00  149614.48    265871.48
2007         48564.30  124557.00    173121.30
2008        104332.38   78622.40    182954.78
2009        164785.64  164666.80    329452.44
2010        215066.50   96809.61    311876.11
2011         50539.39  279293.13    329832.52
2012         43515.34  240235.19    283750.53
2013        132912.35  162118.08    295030.43
2014         96567.00  182362.92    278929.92
```

```
2015            109439.25   158708.96    268148.21
2016             77053.95    73575.86    150629.81
2017             80336.93   108044.62    188381.55
2018            163009.50   112267.55    275277.05
2019             77479.25   128207.25    205686.50
2020             94443.01   101739.55    196182.56
2021             69304.70    50970.00    120274.70
Grand Total    3346740.06  3445290.79   6792030.85
```

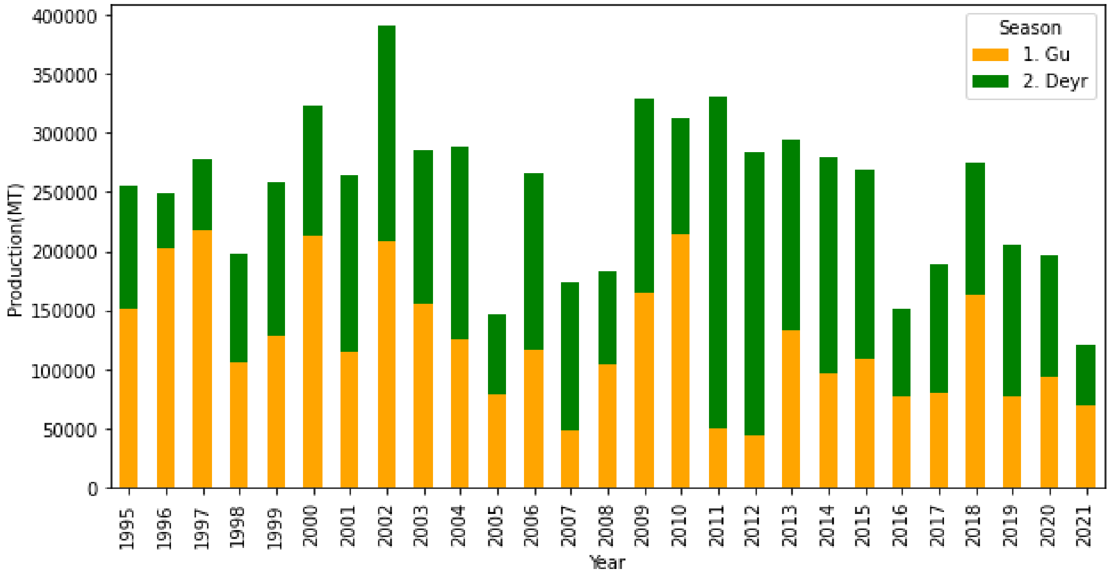
Create another Pivot table without grand totals for plotting

[6]:

```
pivot2 = crop.pivot_table(index = "Year", columns = "Season", values =␣
 ↪"Production",aggfunc = np.sum)
```

[7]:

```
pivot2.plot(kind = 'bar', stacked = True, color = ['orange', 'green'], figsize␣
 ↪= (10, 5), ylabel = 'Production(MT)')
plt.show()
```



[ ]: