

bar_plot

November 27, 2019

1 Create a bar plot in Python

by **Julian Laabs**

This is a small and fast example to create a bar plot with the data structure and data analysis tool of [pandas](#). And it is a first try for me to design a small tutorial in [JupyterLab](#). I assume here that Python, etc. is already installed. If not, check out the Python Distribution [Anaconda](#) and follow the install instructions there. Then you are ready to repeat the following lines of code and you should get the same results.

1.1 Import pandas and load a CSV file

Import the pandas package straightforward and rename it to pd. The only purpose behind the renaming is that one does not need to write the package's full name when calling specific functions.

```
[1]: import pandas as pd
```

As `pandas.read_csv()` will create an own index for an imported dataframe, we specify the index column as the `grab_ID` column of our dataset.

```
[2]: df = pd.read_csv("Zvejnieki/dataset_Zvejnieki.csv", index_col = 'grab_ID')
df.head()
```

```
[2]:
```

	bestattungs_ID	grabnr_literatur	anzahl_bestattungen	laenge_grabgrube	\
grab_ID					
1	1	Grab 1	1	NaN	
2	2	Grab 2	1	1.8	
3	3	Grab 3	1	NaN	
4	4	Grab 4	1	1	
5	5	Grab 5	1	NaN	

	breite_grabgrube	tiefe_grabgrube	\
grab_ID			
1	0.45	0.30	
2	0.40	0.35	
3	0.50	0.40	
4	0.80	0.45	
5	0.30	0.07	

grab_ID	grabbau	stoerung/erhaltung	\
1	NaN	strak gestoert	
2	NaN	NaN	
3	kohleartige Einfuellung	wenig gestoert	
4	kohleartige Einfuellung	strak gestoert	
5	schwarze Kulturerde auf der Grabsohle	teilweise gestoert	

grab_ID	biologisches_geschlecht	alter	... zahn_Kleinsaeuger	zahn_Mensch	\
1	unbest	erwachsen	...	0	0
2	m	erwachsen	...	0	0
3	unbest	erwachsen	...	0	0
4	unbest	erwachsen	...	0	0
5	unbest	infans	...	0	0

grab_ID	zahn_Baer	zahn_unbestimmt	knochen_Vogel	knochen_Bieber	\
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	
5	0	0	0	0	

grab_ID	knochen_unbestimmt	hufe	figur	schildkroetenpanzer
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0

[5 rows x 52 columns]

1.2 Create a barplot

Pandas has inbuild functions to plot data from dataframes ([pandas.DataFrame.plot](#)). But before-hand we need to prepare the data a little more. In our case we would like to count the appearances of the biological sexes of the deceased in our dataset. Therefore, we use the [pandas.Series.value_counts](#) function on the column 'biologisches_geschlecht' to do so.

```
[3]: df['biologisches_geschlecht'].value_counts()
```

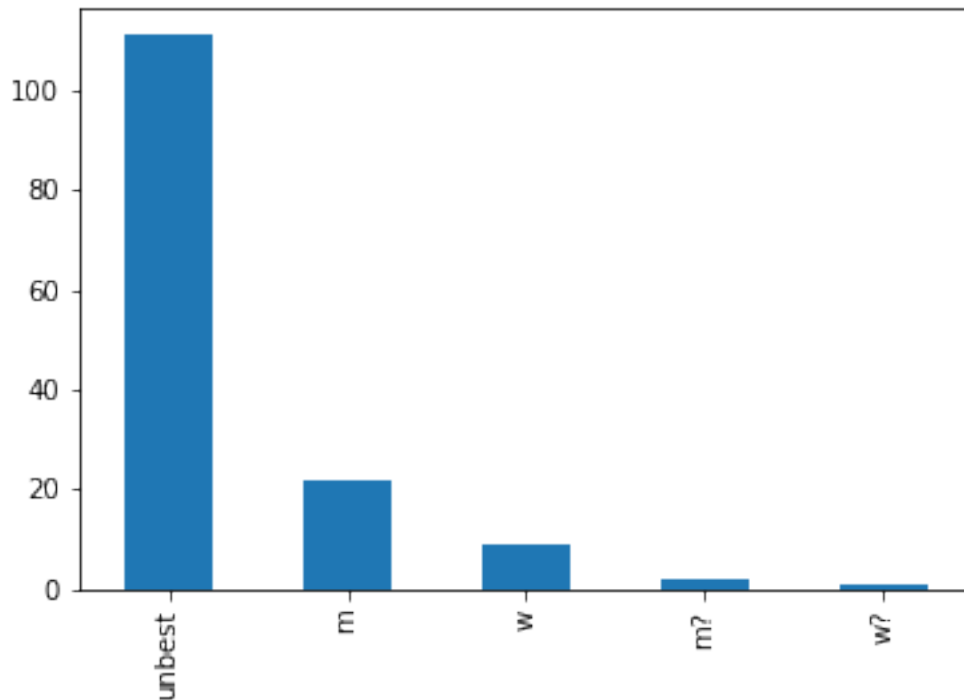
```
[3]: unbest    111
      m         22
      w          9
      m?         2
```

```
w?      1
Name: biologisches_geschlecht, dtype: int64
```

Then we can plot directly.

```
[5]: df['biologisches_geschlecht'].value_counts().plot(kind='bar')
```

```
[5]: <matplotlib.axes._subplots.AxesSubplot at 0x221705c4b08>
```



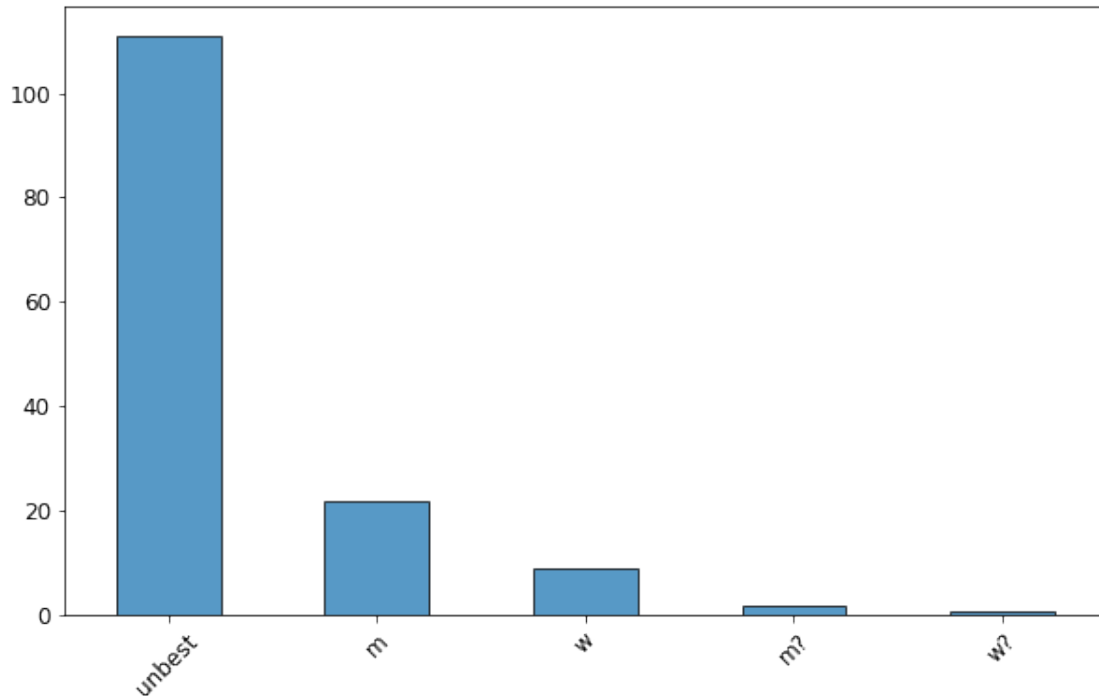
1.3 Plot the data (more respecable)

The `pandas.DataFrame.plot` function has many arguments one can use to change the appearance of the data. We will use only some of them and you can find those an more in this [user guide](#).

- `figsize=(width, hight)` determines the width and heigt of teh fiure in inch
- `edgecolor` will draw a line around the bars, what sets a nice contrast
- `alpha` sets the transparency of the bars and the edgecolor
- `rot` rotates the x-axis labels by a given angle
- `fontsize` de-/increases the size of the x- and y-axis labels

```
[6]: df['biologisches_geschlecht'].value_counts().plot(kind='bar', figsize=(10, 6),
    ↳ edgecolor = 'black', alpha = 0.75, rot = 45.0, fontsize = 12)
```

```
[6]: <matplotlib.axes._subplots.AxesSubplot at 0x22170629648>
```



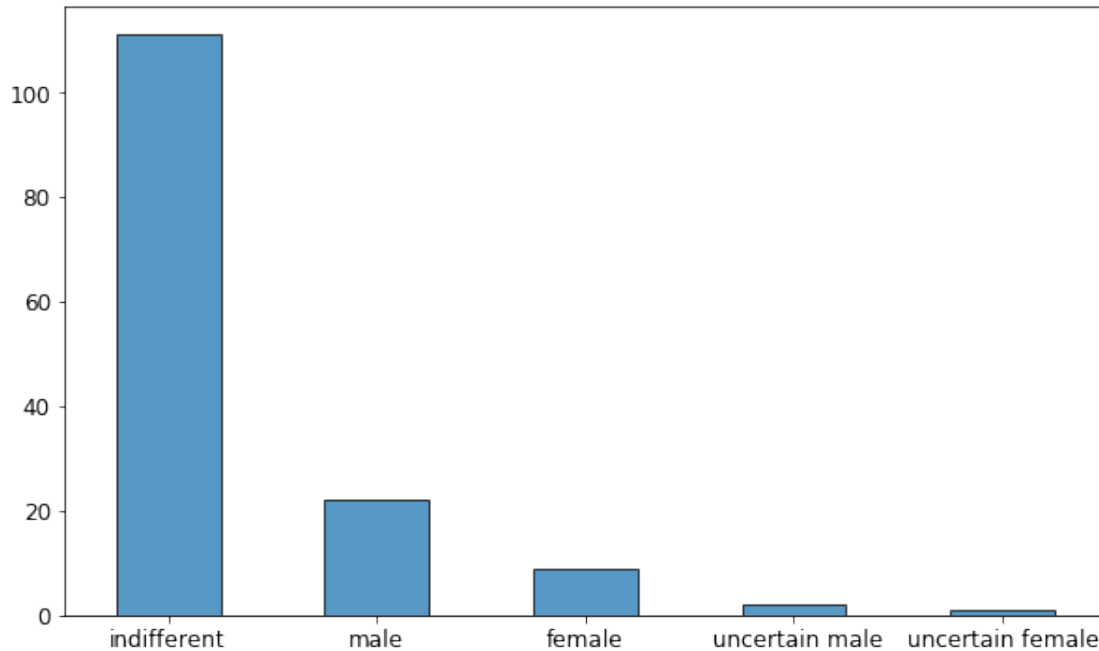
As the labels m, w, etc. are not really meaningful we should change them to proper words. We will there for overwrite specific values of column cells by a new value. This can also be done in the CSV spreadsheet directly.

```
[7]: df.loc[df.biologisches_geschlecht == 'unbest', 'biologisches_geschlecht'] = 'indifferent'
df.loc[df.biologisches_geschlecht == 'm', 'biologisches_geschlecht'] = 'male'
df.loc[df.biologisches_geschlecht == 'w', 'biologisches_geschlecht'] = 'female'
df.loc[df.biologisches_geschlecht == 'm?', 'biologisches_geschlecht'] = 'uncertain male'
df.loc[df.biologisches_geschlecht == 'w?', 'biologisches_geschlecht'] = 'uncertain female'
```

And than plot again.

```
[8]: df['biologisches_geschlecht'].value_counts().plot(kind='bar', figsize=(10, 6),
edgecolor = 'black', alpha = 0.75, rot = 0.0, fontsize = 12)
```

```
[8]: <matplotlib.axes._subplots.AxesSubplot at 0x221706d7548>
```



One last thing, to change the appearance of the plot in a fast way there is a convenient option given by the package `matplotlib`. There are several plot styles which can be chosen from and will change the overall appearance of the plot.

```
[9]: import matplotlib.pyplot as plt

plt.style.available
```

```
[9]: ['bmh',
      'classic',
      'dark_background',
      'fast',
      'fivethirtyeight',
      'ggplot',
      'grayscale',
      'seaborn-bright',
      'seaborn-colorblind',
      'seaborn-dark-palette',
      'seaborn-dark',
      'seaborn-darkgrid',
      'seaborn-deep',
      'seaborn-muted',
      'seaborn-notebook',
      'seaborn-paper',
      'seaborn-pastel',
      'seaborn-poster',
```

```

'seaborn-talk',
'seaborn-ticks',
'seaborn-white',
'seaborn-whitegrid',
'seaborn',
'Solarize_Light2',
'tableau-colorblind10',
'_classic_test']

```

We can call one of those styles and perform the plotting again to see what happens.

```

[10]: plt.style.use('ggplot')

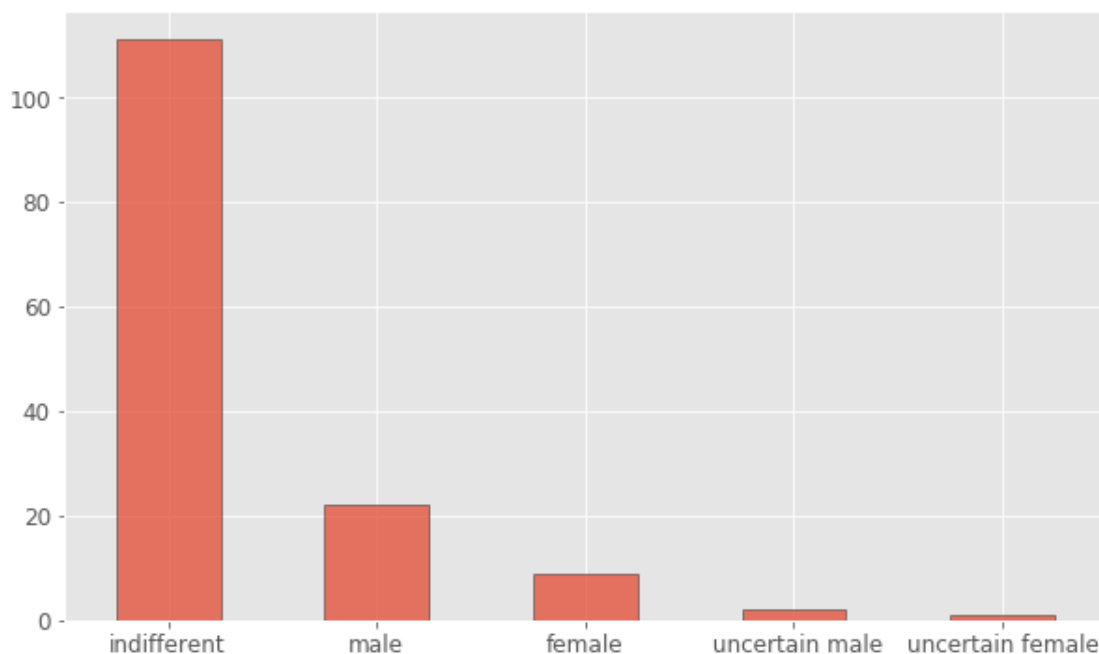
df['biologisches_geschlecht'].value_counts().plot(kind='bar', figsize=(10, 6),
→edgecolor = 'black', alpha = 0.75, rot = 0.0, fontsize = 12)

```

```

[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2217087f848>

```



Or another one.

```

[11]: plt.style.use('dark_background')

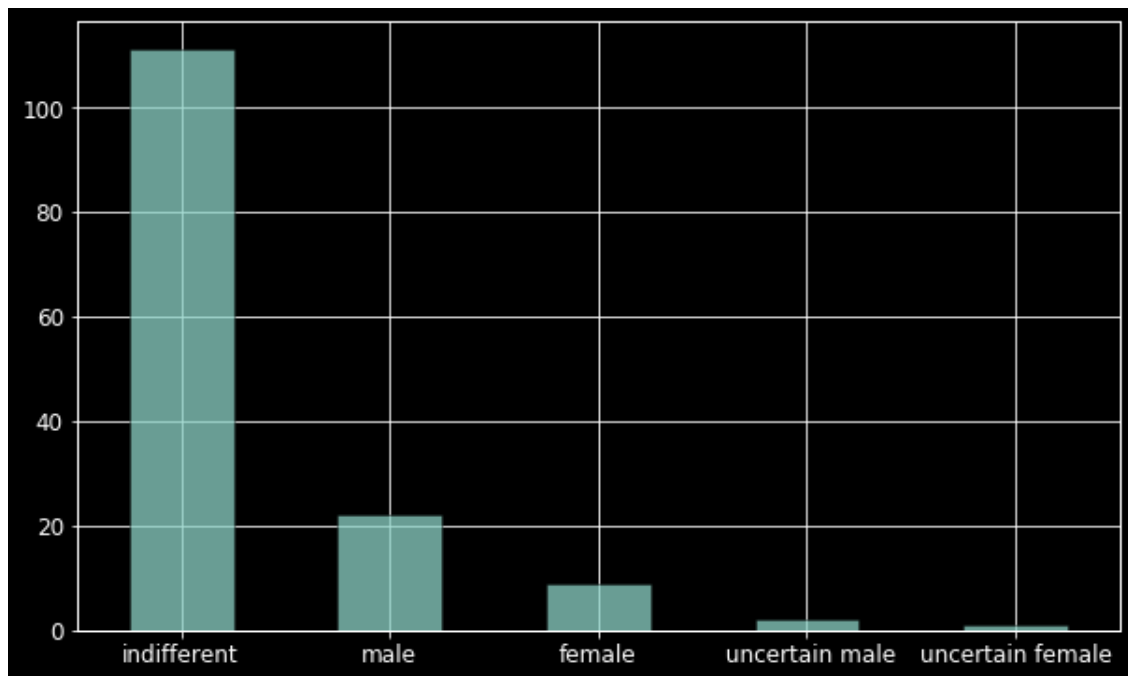
df['biologisches_geschlecht'].value_counts().plot(kind='bar', figsize=(10, 6),
→edgecolor = 'black', alpha = 0.75, rot = 0.0, fontsize = 12)

```

```

[11]: <matplotlib.axes._subplots.AxesSubplot at 0x22170772f88>

```



This should be all. May try creating a bar plot with another column of the data set.