4.1) Tabular Data and Pandas

Vitor Kamada

December 2019

Reference

Tables, Graphics, and Figures from

Principles and Techniques of Data Science

Lau et al. (2019): Ch 3 Tabular Data and pandas

`https://www.textbook.ds100.org/ch/03/pandas_intro.html`

US Social Security: Baby Names

```
import pandas as pd
path ='https://github.com/DS-100/textbook/raw/master/content/ch/'
baby = pd.read_csv(path + '03/babynames.csv')
```

	Name	Sex	Count	Year
0	Mary	F	9217	1884
1	Anna	F	3860	1884
2	Emma	F	2587	1884

```
baby.loc[1, 'Name']
```

```
'Anna'
```

```
baby.loc[1:5, 'Name':'Count']
```

	Name	Sex	Count
1	Anna	F	3860
2	Emma	F	2587
3	Elizabeth	F	2549
4	Minnie	F	2243
5	Margaret	F	2142

```
baby.loc[:, ['Name', 'Year']]
```

	Name	Year
0	Mary	1884
1	Anna	1884
2	Emma	1884

```
baby['Year'] == 2016
```

```
0        False
1        False
2        False
```

```
baby_2016 = baby.loc[baby['Year'] == 2016, :]
```

	Name	Sex	Count	Year
1850880	Emma	F	19414	2016
1850881	Olivia	F	19246	2016
1850882	Ava	F	16237	2016

```
sorted_2016 = baby_2016.sort_values('Count', ascending=False)
```

	Name	Sex	Count	Year
1850880	Emma	F	19414	2016
1850881	Olivia	F	19246	2016
1869637	Noah	M	19015	2016

```
sorted_2016.iloc[0, 0]
```

```
'Emma'
```

```
year_counts = baby[['Year', 'Count']].groupby('Year').count()
```

```
grouped_counts = baby.groupby(['Year', 'Sex']).sum()
```

Year	Count
1880	2000
1881	1935
1882	2127

Year	Sex	Count
1880	F	90992
	M	110491
1881	F	91953
	M	100743

```python
def most_popular(series):
    return series.iloc[0]

baby_pop = baby.groupby(['Year', 'Sex']).agg(most_popular)
```

		Name	Count
Year	**Sex**		
1880	**F**	Mary	7065
	M	John	9655
1881	**F**	Mary	6919
	M	John	8769

```python
baby_pop.loc[(2000, 'F'), 'Name']  'Emily'
```

```
pd.pivot_table(baby,
               index='Year',
               columns='Sex',
               values='Name',
               aggfunc=most_popular)
```

Sex	F	M
Year		
1880	Mary	John
1881	Mary	John
1882	Mary	John
1883	Mary	John

```
names = baby['Name']
names.apply(len)
```

```
0    4
1    4
2    4
3    9
```

```
def last_letter(string):
    return string[-1]

names.apply(last_letter)
```

```
0    y
1    a
2    a
3    h
```

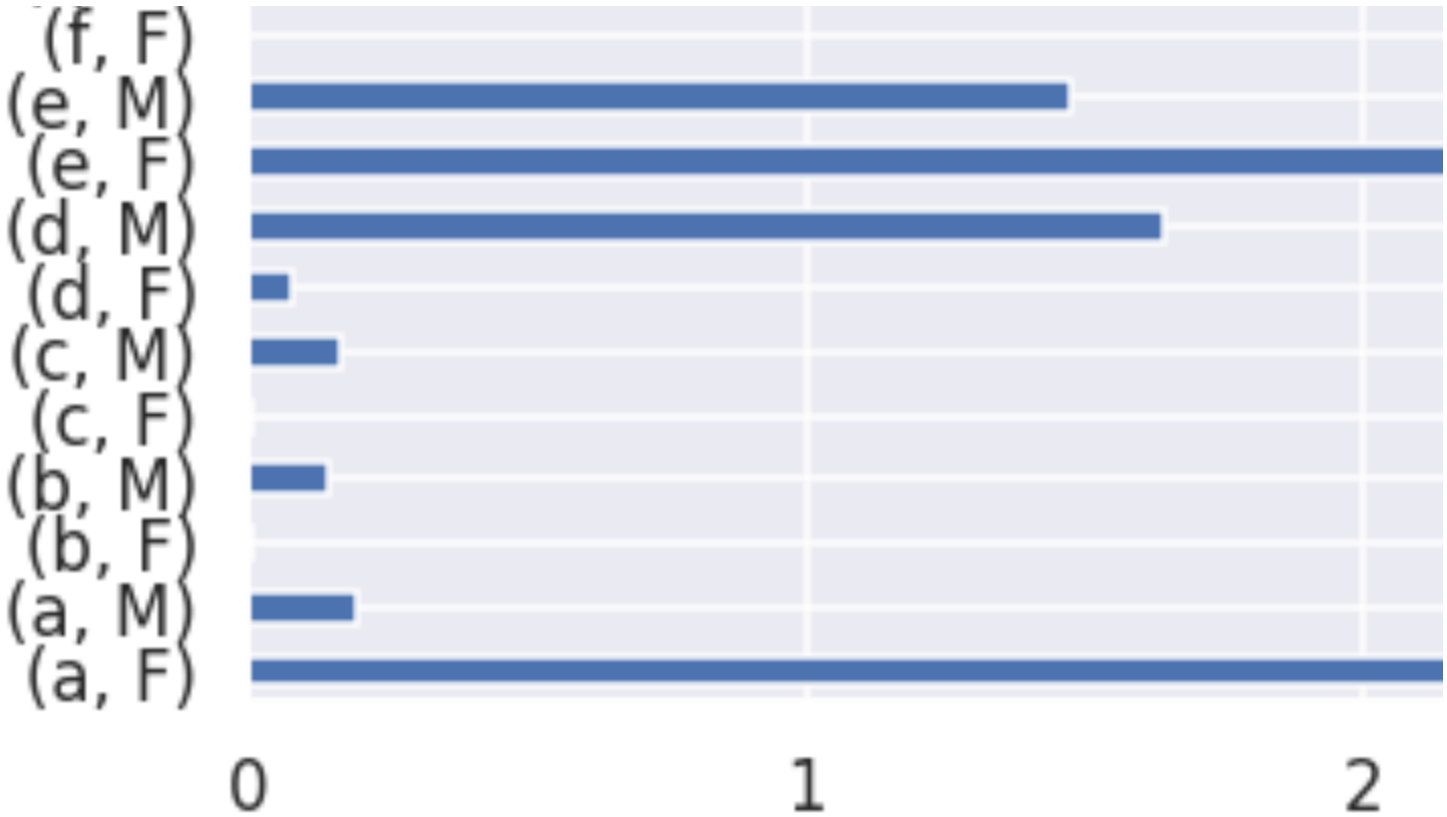
```
baby['Last'] = names.str[-1]
```

```
letter_dist = (baby[['Last', 'Sex', 'Count']]
    .groupby(['Last', 'Sex']).sum())
```

	Name	Sex	Count	Year	Last
0	Mary	F	9217	1884	y
1	Anna	F	3860	1884	a
2	Emma	F	2587	1884	a

		Count
Last	**Sex**	
a	**F**	58079486
	M	1931630
b	**F**	17376
	M	1435939

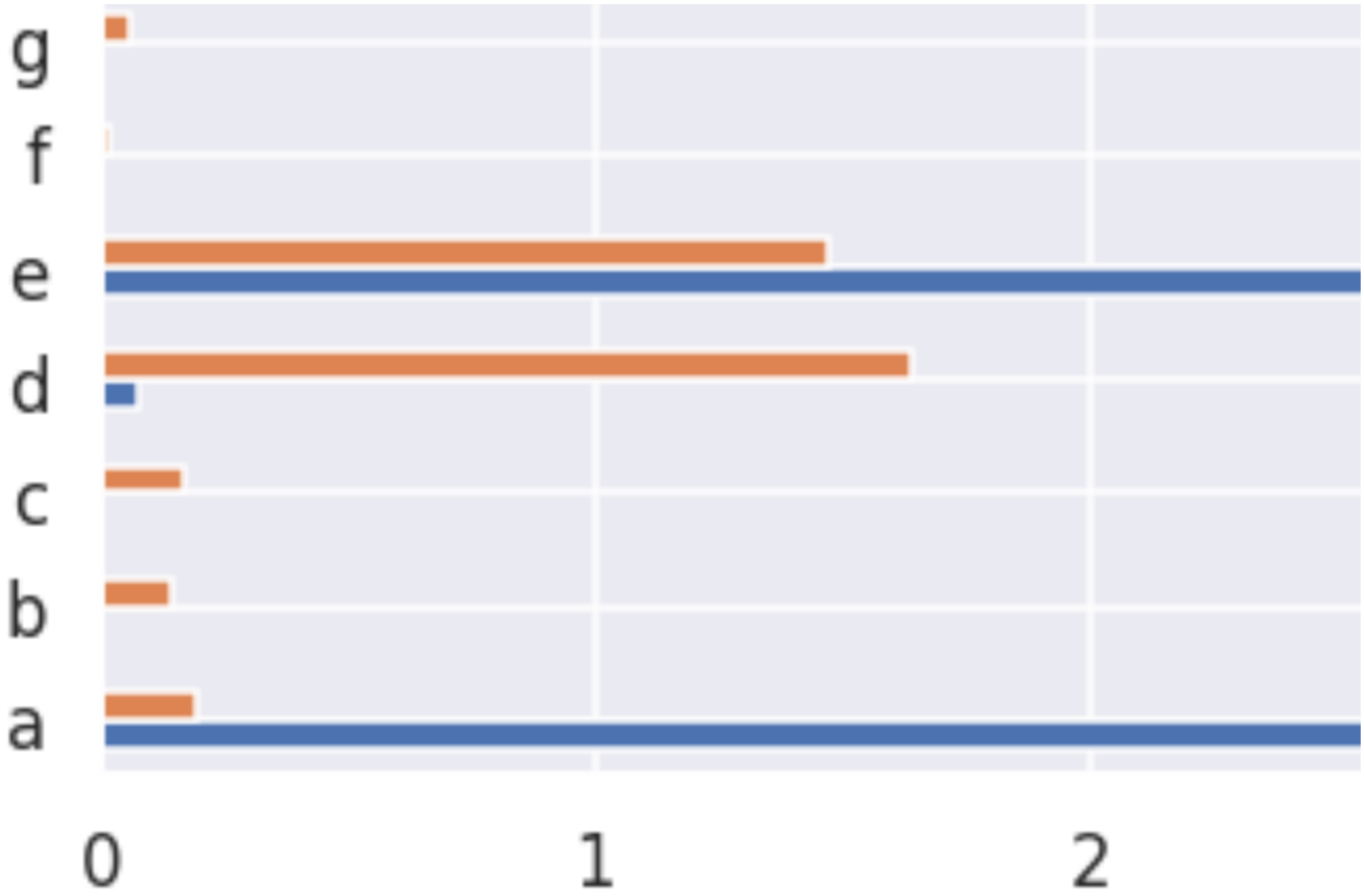
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set()
sns.set_context('talk')
letter_dist.plot.barh(figsize=(20, 20))
```



```
letter_pivot = pd.pivot_table(baby, index='Last',
    columns='Sex', values='Count', aggfunc='sum')
```

Sex	F	M
Last		
a	58079486	1931630
b	17376	1435939
c	30262	1672407

```
letter_pivot.plot.barh(figsize=(13, 13))
```



```
total_for_each_letter = letter_pivot['F'] + letter_pivot['M']
letter_pivot['F prop'] = letter_pivot['F'] / total_for_each_letter
letter_pivot['M prop'] = letter_pivot['M'] / total_for_each_letter
```

Sex	F	M	F prop	M prop
Last				
a	58079486	1931630	0.967812	0.032188
b	17376	1435939	0.011956	0.988044
c	30262	1672407	0.017773	0.982227

```
(letter_pivot[['F prop', 'M prop']].sort_values('M prop')
  .plot.barh(figsize=(10, 10)))
```