`pandas` 의 자료형에는 `숫자(int, float)` , `문자(object, category)` , `날짜(date)` 가 있다.

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

실습을 위해서 만만한 `tips` 데이터를 불러온다.

In [2]:
```
tips = sns.load_dataset('tips')
```

`info` 를 통해 살펴보면 `memory usage` 라는 것이 있다. `memory useage` 란, 메모리 상에 데이터를 올리고 큰 데이터의 경우 분산처리를 하여 올린다.

In [3]:
```
tips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
total_bill     244 non-null float64
tip            244 non-null float64
sex            244 non-null category
smoker         244 non-null category
day            244 non-null category
time           244 non-null category
size           244 non-null int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.2 KB
```

`corr` 메소드는 상관분석(correlation)을 의미한다.

In [4]:
```
tips.corr()
```

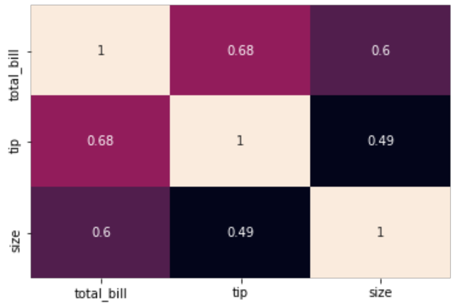
Out[4]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

이에 대한 `heatmap` 을 그려본다.

In [5]:
```
sns.heatmap(tips.corr(), cbar=False, annot=True)
```

Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71ccaf72b0>
```



흡연자 중에서 남자와 여자의 수를 알고 싶을 때, 다음과 같이 `value_counts` 를 사용한다.

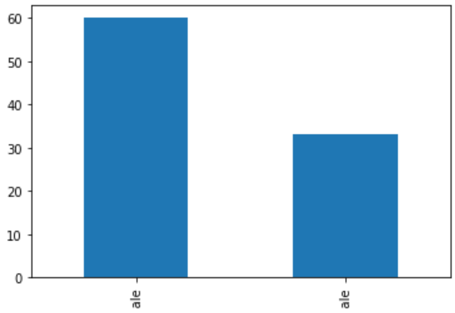
In [6]:
```
tips[tips.smoker=='Yes'].sex.value_counts()
```

Out[6]:
```
Male      60
Female    33
Name: sex, dtype: int64
```

이에 대한 그래프를 그려보면 다음과 같다.

In [7]:
```
tips[tips.smoker=='Yes'].sex.value_counts().plot.bar()
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71c46b42b0>
```



`팬시 인덱싱`으로 다음과 같이 뽑아올 수도 있다.

In [8]:
```
tips[tips.smoker=='Yes'][['sex','smoker']].sample(5)
```

Out[8]:

	sex	smoker
202	Female	Yes
189	Male	Yes
199	Male	Yes
191	Female	Yes
213	Female	Yes

`iteritems`를 통해 제너레이터를 생성하여 `next` 함수로 순회할 수 있다. `DataFrame` 객체의 `iteritems`를 불러오면 각 컬럼에 대한 열(row)을 가져온다.

In [9]:
```
iter_ = tips.iteritems()
```

In [10]:
```
next_ = next(iter_)
```

In [11]:
```
next_[0]
```

Out[11]:
```
'total_bill'
```

In [12]:
```
next_[1][:10]
```

Out[12]:
```
0    16.99
1    10.34
2    21.01
3    23.68
4    24.59
5    25.29
6     8.77
7    26.88
8    15.04
9    14.78
Name: total_bill, dtype: float64
```

행으로 가져오고 싶을 때는 `iterrows`로 가져올 수 있다.

In [13]:
```
next(tips.iterrows())
```

Out[13]:
```
(0, total_bill     16.99
 tip             1.01
 sex           Female
 smoker            No
 day              Sun
 time          Dinner
 size               2
 Name: 0, dtype: object)
```

vincent

In [14]:
```
# !pip install vincent
```

In [15]:
```
# !pip install -q pdvega
```

In [16]:
```
tips[['total_bill','smoker']].set_index('smoker')
```

Out[16]:

	total_bill
smoker	
No	16.99
No	10.34
No	21.01
No	23.68
No	24.59
No	25.29
No	8.77
No	26.88
No	15.04
No	14.78
No	10.27
No	35.26
No	15.42
No	18.43
No	14.83
No	21.58
No	10.33

	total_bill
smoker	
No	16.29
No	16.97
No	20.65
No	17.92
No	20.29
No	15.77
No	39.42
No	19.82
No	17.81
No	13.37
No	12.69
No	21.70
No	19.65
...	...
Yes	28.17
Yes	12.90
Yes	28.15
Yes	11.59
Yes	7.74
Yes	30.14
Yes	12.16
Yes	13.42
Yes	8.58
No	15.98
Yes	13.42
Yes	16.27
Yes	10.09
No	20.45
No	13.28
Yes	22.12
Yes	24.01
Yes	15.69
No	11.61
No	10.77
Yes	15.53
No	10.07
Yes	12.60
Yes	32.83
No	35.83
No	29.03
Yes	27.18
Yes	22.67
No	17.82
No	18.78

244 rows × 1 columns

```
In [17]: x = tips[['total_bill','smoker']].groupby('smoker')
         x.mean()
```

Out[17]:

	total_bill
smoker	
Yes	20.756344
No	19.188278

```
In [18]: s = tips.groupby('smoker').mean().total_bill
```

pdvega

```
In [19]: # !pip install --upgrade pdvega
```

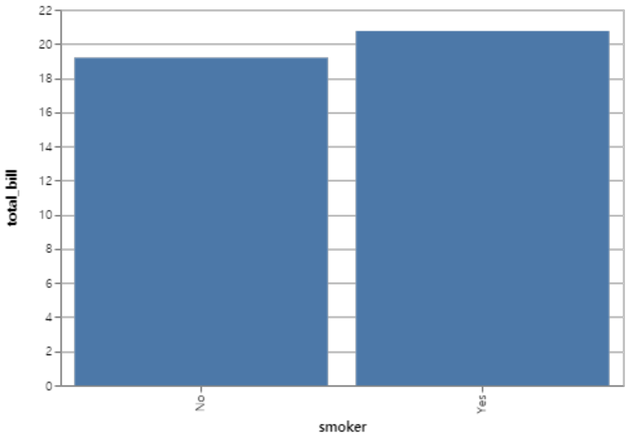
```
In [20]: import pdvega
```

아래는 에러 메시지가 거슬려서 넣어주었다.

```
In [21]: import warnings
         warnings.filterwarnings('ignore')
```

```
In [22]: s.vgplot.bar()
```

In [22]:
```
s.vgplot.bar()
```



In [23]:
```
tips.pivot_table(index='smoker',columns='sex', aggfunc=np.sum, margins=True)
```

Out[23]:

	size			tip			total_bill		
sex	**Male**	**Female**	**All**	**Male**	**Female**	**All**	**Male**	**Female**	**All**
smoker									
Yes	150	74	224	183.07	96.74	279.81	1337.07	593.27	1930.34
No	263	140	403	302.00	149.77	451.77	1919.75	977.68	2897.43
All	413	214	627	485.07	246.51	731.58	3256.82	1570.95	4827.77

In [24]:
```
pd.crosstab([tips.smoker, tips.sex], tips.time, values=tips.tip, aggfunc=np.mean)
```

Out[24]:

	time	**Lunch**	**Dinner**
smoker	**sex**		
Yes	**Male**	2.790769	3.123191
	Female	2.891000	2.949130
No	**Male**	2.941500	3.158052
	Female	2.459600	3.044138

In [25]:
```
pd.crosstab([tips.smoker, tips.sex], tips.time, values=tips.tip, aggfunc=np.mean).index
```

Out[25]:
```
MultiIndex(levels=[['Yes', 'No'], ['Male', 'Female']],
           codes=[[0, 0, 1, 1], [0, 1, 0, 1]],
           names=['smoker', 'sex'])
```

reset index

In [26]:
```
x = tips[tips.sex=='Male'].loc[:15]
```

In [27]:
```
x.reset_index(drop=True)
```

Out[27]:

	total_bill	tip	sex	smoker	day	time	size
0	10.34	1.66	Male	No	Sun	Dinner	3
1	21.01	3.50	Male	No	Sun	Dinner	3
2	23.68	3.31	Male	No	Sun	Dinner	2
3	25.29	4.71	Male	No	Sun	Dinner	4
4	8.77	2.00	Male	No	Sun	Dinner	2
5	26.88	3.12	Male	No	Sun	Dinner	4
6	15.04	1.96	Male	No	Sun	Dinner	2
7	14.78	3.23	Male	No	Sun	Dinner	2
8	10.27	1.71	Male	No	Sun	Dinner	2
9	15.42	1.57	Male	No	Sun	Dinner	2
10	18.43	3.00	Male	No	Sun	Dinner	4
11	21.58	3.92	Male	No	Sun	Dinner	2

In [28]:
```
tips.groupby(['sex','smoker']).mean()[['tip']]
```

Out[28]:

		tip
sex	**smoker**	
Male	**Yes**	3.051167
	No	3.113402
Female	**Yes**	2.931515
	No	2.773519

stack and unstack

```
In [29]: group = tips.groupby(['sex','smoker']).mean()
```

```
In [30]: group[['tip']].unstack()
```

Out[30]:

	tip	
smoker	Yes	No
sex		
Male	3.051167	3.113402
Female	2.931515	2.773519

```
In [31]: group[['tip']].stack()
```

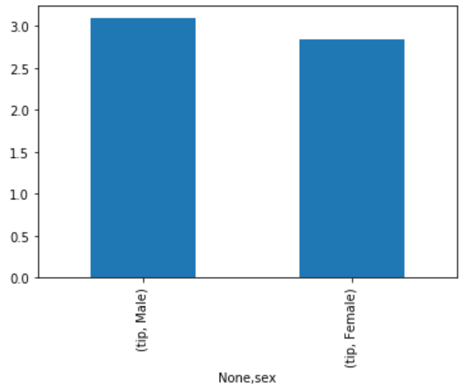
```
Out[31]: sex     smoker
         Male    Yes     tip    3.051167
                 No      tip    3.113402
         Female  Yes     tip    2.931515
                 No      tip    2.773519
         dtype: float64
```

```
In [32]: tips.groupby('sex').mean()[['tip']].unstack()
```

```
Out[32]:      sex
         tip  Male      3.089618
              Female    2.833448
         dtype: float64
```

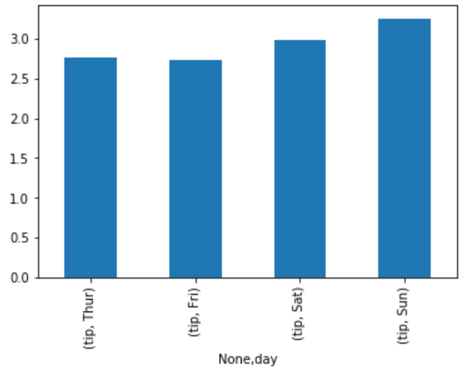
```
In [33]: tips.groupby('sex').mean()[['tip']].unstack().plot.bar()
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7f71c462a320>
```



```
In [34]: tips.groupby('day').mean()[['tip']].unstack().plot.bar(stacked=False)
```

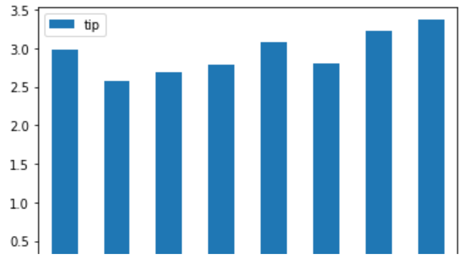
```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f71c457acf8>
```

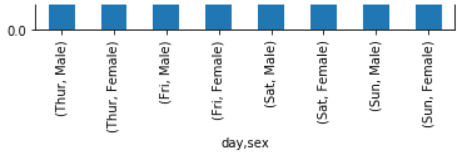


```
In [35]: group = tips.groupby(['day','sex']).mean()
```

```
In [36]: group[['tip']].plot.bar()
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f71c4562438>
```





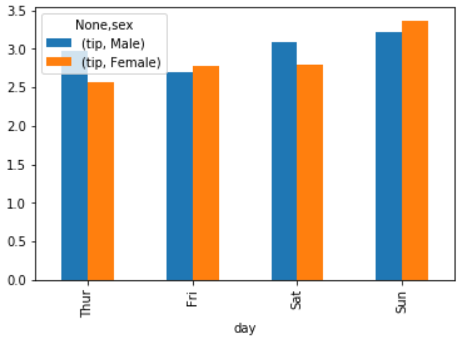
In [37]:
```
group[['tip']].unstack()
```

Out[37]:

	tip	
sex	**Male**	**Female**
day		
Thur	2.980333	2.575625
Fri	2.693000	2.781111
Sat	3.083898	2.801786
Sun	3.220345	3.367222

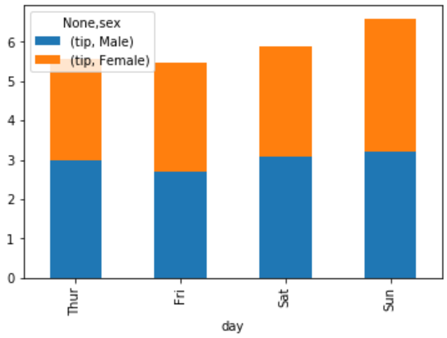
In [38]:
```
group[['tip']].unstack().plot.bar()
```

Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71c46a2dd8>
```



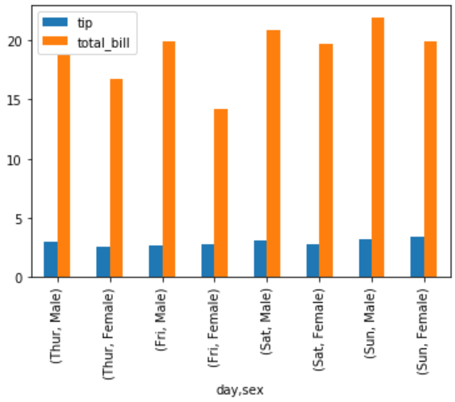
In [39]:
```
group[['tip']].unstack().plot.bar(stacked=True)
```

Out[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71c4460198>
```



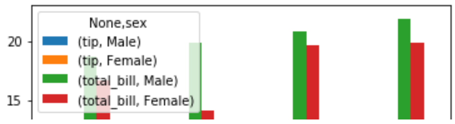
In [40]:
```
group[['tip','total_bill']].plot.bar()
```

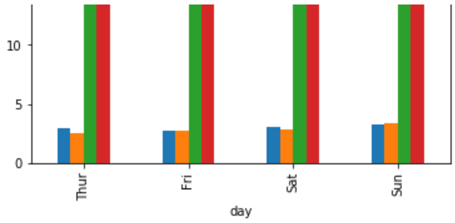
Out[40]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71c43e3c88>
```



In [41]:
```
group[['tip','total_bill']].unstack().plot.bar()
```

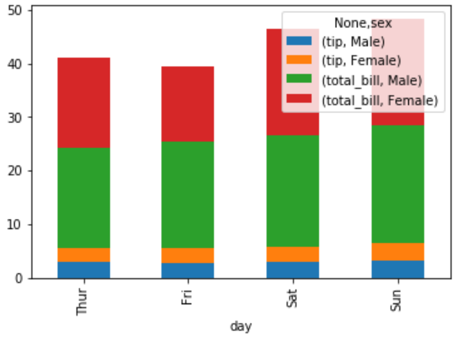
Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71c43660f0>
```





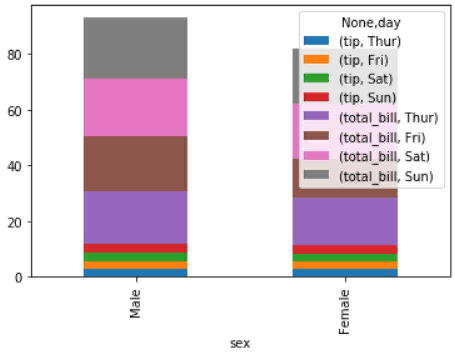
```
In [42]: group[['tip','total_bill']].unstack().plot.bar(stacked=True)
```

```
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7f71c42e64e0>
```



```
In [43]: group[['tip','total_bill']].unstack(0).plot.bar(stacked=True)
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f71c41ffd30>
```



IPA 주관 인공지능센터 기본(fundamental) 과정

- GitHub link: here
- E-Mail: windkyle7@gmail.com