

① drinks.groupby('continent').beer\_servings.mean()

② drinks.groupby('continent').mean()

③ drinks.groupby('continent').median()

④ .groupby('con').split-sec.agg(['mean', 'min', 'max'])

⑤ .agg(['mean', 'min', 'max'])

def fn():  
if x == 'm':  
return 1

df = pd.DataFrame(  
{"a": [1, 2],  
"b": [2, 3],  
"index": [1, 2]}

	a	b
1	1	2
2	2	3

① df.iloc[0,0] ←

④

② df.loc[1, 'a'] ←

④

③ df.loc[df['a'] > 1] ←

	a	b
2	2	3

④ df.loc[(df['a'] > 0) & (df['b'] < 3)]

3

users.groupby('gender').apply(gender-to-numeric)

a = users.groupby('occupation').gender\_n.sum() /  $\frac{\text{value-count}}{100}$

a.sort\_values(ascending=False)

⑥ users.groupby(['s', 'c']).age.mean()

Filtering Operations

newdf = df[(df.origin == "JK") & (df.carrier == "86")]

newdf = df.query("origin == 'JK' & carrier == '86'")

loc & iloc (diff) loc considers rows based on index labels  
wheras iloc " " based on position so it only takes integers

iloc (Position)

df.iloc[0:5]

(Selecting rows based on index or row position)

loc (index label)

df.loc['a':'c']

col 1  
a 1  
b 2  
c 3