Pandas扩展知识

1. 加载数据

```
# https://gist.github.com/tijptjik/9408623 wine.csv
import pandas as pd
data = pd.read_csv("wine.csv")
```

```
#data
data.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Proanth	Color.int	Hue
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04

```
print(data.head())
print(data.tail(3))
print(data.describe())
print("***")
print(data.columns)
```

```
   Wine  Alcohol  Malic.acid   Ash   Acl   Mg  Phenols  Flavanoids  \
0     1    14.23        1.71  2.43  15.6  127     2.80        3.06   
1     1    13.20        1.78  2.14  11.2  100     2.65        2.76   
2     1    13.16        2.36  2.67  18.6  101     2.80        3.24   
3     1    14.37        1.95  2.50  16.8  113     3.85        3.49   
4     1    13.24        2.59  2.87  21.0  118     2.80        2.69   

   Nonflavanoid.phenols  Proanth  Color.int   Hue    OD  Proline  
0                  0.28     2.29       5.64  1.04  3.92     1065  
1                  0.26     1.28       4.38  1.05  3.40     1050  
2                  0.30     2.81       5.68  1.03  3.17     1185  
3                  0.24     2.18       7.80  0.86  3.45     1480  
4                  0.39     1.82       4.32  1.04  2.93      735  
     Wine  Alcohol  Malic.acid   Ash   Acl   Mg  Phenols  Flavanoids  \
175     3    13.27        4.28  2.26  20.0  120     1.59        0.69   
176     3    13.17        2.59  2.37  20.0  120     1.65        0.68   
177     3    14.13        4.10  2.74  24.5   96     2.05        0.76   

     Nonflavanoid.phenols  Proanth  Color.int   Hue    OD  Proline  
175                  0.43     1.35       10.2  0.59  1.56      835  
176                  0.53     1.46        9.3  0.60  1.62      840  
177                  0.56     1.35        9.2  0.61  1.60      560  
             Wine     Alcohol  Malic.acid         Ash         Acl          Mg  \
count  178.000000  178.000000  178.000000  178.000000  178.000000  178.000000   
mean     1.938202   13.000618    2.336348    2.366517   19.494944   99.741573   
std      0.775035    0.811827    1.117146    0.274344    3.339564   14.282484   
min      1.000000   11.030000    0.740000    1.360000   10.600000   70.000000   
25%      1.000000   12.362500    1.602500    2.210000   17.200000   88.000000   
50%      2.000000   13.050000    1.865000    2.360000   19.500000   98.000000   
75%      3.000000   13.677500    3.082500    2.557500   21.500000  107.000000   
max      3.000000   14.830000    5.800000    3.230000   30.000000  162.000000   
```

```
         Phenols  Flavanoids  Nonflavanoid.phenols     Proanth    Color.int  \
count  178.000000  178.000000            178.000000  178.000000  178.000000   
mean     2.295112    2.029270              0.361854    1.590899    5.058090   
std      0.625851    0.998859              0.124453    0.572359    2.318286   
min      0.980000    0.340000              0.130000    0.410000    1.280000   
25%      1.742500    1.205000              0.270000    1.250000    3.220000   
50%      2.355000    2.135000              0.340000    1.555000    4.690000   
75%      2.800000    2.875000              0.437500    1.950000    6.200000   
max      3.880000    5.080000              0.660000    3.580000   13.000000   

             Hue          OD      Proline  
count  178.000000  178.000000   178.000000  
mean     0.957449    2.611685   746.893258  
std      0.228572    0.709990   314.907474  
min      0.480000    1.270000   278.000000  
25%      0.782500    1.937500   500.500000  
50%      0.965000    2.780000   673.500000  
75%      1.120000    3.170000   985.000000  
max      1.710000    4.000000  1680.000000  
"""
Index(['Wine', 'Alcohol', 'Malic.acid', 'Ash', 'Acl', 'Mg', 'Phenols',
       'Flavanoids', 'Nonflavanoid.phenols', 'Proanth', 'Color.int', 'Hue',
       'OD', 'Proline'],
      dtype='object')
```

```
data.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Proanth	Color.int	Hue	OD	Proline
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

2. Pandas中的统计和汇总方法

我们可以加载自己的数据集到panda的DataFrame. 以此为例.

```
import pandas as pd

df = pd.read_csv("mydataset.csv")

print(df.head())
print(df.tail(3))
# 统计方法: describe() 针对Series和DataFrame的各列计算汇总统计
print(df.describe())
print("各列的最小值/最大值:")
print(df.min())
print(df.max())
# 求值的总和: 各列的总和
print(df.sum())
print("各列的平均值:")
print(df.mean())
print("各列的方差:")
print(df.var())
print("各列的标准差:")
print(df.std())
print("计算样本的累计和:")
print(df.iloc[:,1:].cumsum())
```

```python
print("计算一阶差分:")
print(df.iloc[:,1:].diff())
# 上面遇到NaN,可以用fillna()填充缺失数据, 也可以用dropna()丢弃含有缺失值的行
print("计算一阶差分:")
print(df.iloc[:,1:].diff().fillna(0))
print("计算一阶差分:")
print(df.iloc[:,1:].diff().fillna(method='bfill'))
print("计算一阶差分:")
print(df.iloc[:,1:].diff().dropna())
print("***")
print(df.columns)
```

```
  Name    Hight  Weight  score
0   Gang     91      95     93
1  Shuan     88      88     88
2   Yang     69      80     70
3     Li     78      79     79
4    Shi     55      88     78
  Name    Hight  Weight  score
2   Yang     69      80     70
3     Li     78      79     79
4    Shi     55      88     78
           Hight     Weight      score
count   5.000000   5.000000   5.000000
mean   76.200000  86.000000  81.600000
std    14.686729   6.595453   9.016651
min    55.000000  79.000000  70.000000
25%    69.000000  80.000000  78.000000
50%    78.000000  88.000000  79.000000
75%    88.000000  88.000000  88.000000
max    91.000000  95.000000  93.000000
各列的最小值/最大值:
Name      Gang
Hight       55
Weight      79
score       70
dtype: object
Name      Yang
Hight       91
Weight      95
score       93
dtype: object
Name      GangShuanYangLiShi
Hight                    381
Weight                   430
score                    408
dtype: object
各列的平均值:
Hight     76.2
Weight    86.0
score     81.6
dtype: float64
各列的方差:
Hight     215.7
Weight     43.5
score      81.3
dtype: float64
各列的标准差:
Hight     14.686729
Weight     6.595453
score      9.016651
dtype: float64
计算样本的累计和:
   Hight  Weight  score
0     91      95     93
1    179     183    181
2    248     263    251
3    326     342    330
4    381     430    408
计算一阶差分:
   Hight  Weight  score
0    NaN     NaN    NaN
1   -3.0    -7.0   -5.0
2  -19.0    -8.0  -18.0
3    9.0    -1.0    9.0
4  -23.0     9.0   -1.0
计算一阶差分:
   Hight  Weight  score
0    0.0     0.0    0.0
```

```
1   -3.0    -7.0   -5.0
2  -19.0    -8.0  -18.0
3    9.0    -1.0    9.0
4  -23.0     9.0   -1.0
计算一阶差分：
   Hight  Weight  score
0   -3.0    -7.0   -5.0
1   -3.0    -7.0   -5.0
2  -19.0    -8.0  -18.0
3    9.0    -1.0    9.0
4  -23.0     9.0   -1.0
计算一阶差分：
   Hight  Weight  score
1   -3.0    -7.0   -5.0
2  -19.0    -8.0  -18.0
3    9.0    -1.0    9.0
4  -23.0     9.0   -1.0
***
Index(['Name  ', 'Hight', 'Weight', 'score'], dtype='object')
```

3. 列索引

```
pwd
```

```
'C:\\Users\\huang\\algorithms_2nd_edition\\lec_research\\kNN_Classification'
```

```python
import pandas as pd
import os

# 获取当前工作的文件夹名.(pwd)
# file = os.path.dirname(__file__)
#print(file)

file = 'C:\\Users\\huang\\algorithms_2nd_edition\\lec_research\\kNN_Classification'
# use read_csv to read data in the dataset as a data frame
#df = pd.read_csv(file+"/DATA/Module2/Datasets/direct_marketing.csv")
df = pd.read_csv(r'C:\\Users\\huang\\algorithms_2nd_edition\\lec_research\\kNN_Classification\\DATA\\Module2\\Datasets\\direct_marketing.csv')

print(df.tail(2))
print("******")

#print(df.recency)
print(df.recency.head(2))

print(df['recency'].head(2))

print("******")
print(df.loc[:,'recency'].head(2))

# 数值索引: pandas中的“iloc”通过数来选择行和列
print(df.iloc[:, 0].head(2))
```

```
       recency history_segment  history  mens  womens   zip_code  newbie  \
63998        1  5) $500 - $750   552.94     1       0  Surburban       1
63999        1  4) $350 - $500   472.82     0       1  Surburban       0

            channel        segment  visit  conversion  spend  DM_category
63998  Multichannel  Womens E-Mail      0           0    0.0            4
63999           Web    Mens E-Mail      0           0    0.0            3
******
0    10
1     6
Name: recency, dtype: int64
0    10
1     6
Name: recency, dtype: int64
******
0    10
1     6
Name: recency, dtype: int64
0    10
1     6
```

```
Name: recency, dtype: int64
```

4. 行索引

```python
# 当前工作的文件夹
file = 'C:\\Users\\huang\\algorithms_2nd_edition\\lec_research\\kNN_Classification'

# use read_csv to read data in the dataset as a data frame
df = pd.read_csv(file+"\\DATA\\Module2\\Datasets\\direct_marketing.csv")
# 行索引
print(df[0:3])
print("******索引: 头两行,所有列***")
print(df.iloc[0:2, :])
print("******索引: 头两行,头5列***")
print(df.iloc[0:2, :5])

print("******索引: 头两行,自选3列***")
print(df.iloc[0:2, [0,1,3]])
```

```
   recency history_segment  history  mens  womens   zip_code  newbie channel  \
0       10  2) $100 - $200   142.44     1       0  Surburban       0   Phone
1        6  3) $200 - $350   329.08     1       1      Rural       1     Web
2        7  2) $100 - $200   180.65     0       1  Surburban       1     Web

         segment  visit  conversion  spend  DM_category
0  Womens E-Mail      0           0    0.0            4
1      No E-Mail      0           0    0.0           11
2  Womens E-Mail      0           0    0.0            1
******索引: 头两行,所有列***
   recency history_segment  history  mens  womens   zip_code  newbie channel  \
0       10  2) $100 - $200   142.44     1       0  Surburban       0   Phone
1        6  3) $200 - $350   329.08     1       1      Rural       1     Web

         segment  visit  conversion  spend  DM_category
0  Womens E-Mail      0           0    0.0            4
1      No E-Mail      0           0    0.0           11
******索引: 头两行,头5列***
   recency history_segment  history  mens  womens
0       10  2) $100 - $200   142.44     1       0
1        6  3) $200 - $350   329.08     1       1
******索引: 头两行,自选3列***
   recency history_segment  mens
0       10  2) $100 - $200     1
1        6  3) $200 - $350     1
```

5. 布尔型索引

```python
# 布尔型索引 (boolean index)
print(df.recency < 7)

#feed back boolean series to regular df
print("feed back boolean series to regular dataframe")
print(df[df.recency<7])

# 将多个布尔型索引组合
print("combine multiple boolean indexing conditions")
print(df[(df.recency<7)&(df.newbie==0)])
```

```
0        False
1         True
2        False
3        False
4         True
5         True
6        False
7        False
8        False
9        False
10       False
11        True
12        True
13        True
14        True
15        True
16        True
17       False
```

```
18       False
19        True
20       False
21       False
22        True
23        True
24        True
25        True
26       False
27        True
28       False
29        True
         ...
63970     True
63971     True
63972     True
63973    False
63974    False
63975    False
63976     True
63977    False
63978    False
63979    False
63980     True
63981     True
63982     True
63983     True
63984     True
63985    False
63986    False
63987     True
63988     True
63989    False
63990     True
63991     True
63992     True
63993     True
63994    False
63995    False
63996     True
63997     True
63998     True
63999     True
Name: recency, Length: 64000, dtype: bool
feed back boolean series to regular dataframe
       recency   history_segment  history  mens  womens   zip_code  newbie  \
1            6    3) $200 - $350   329.08     1       1      Rural       1
4            2      1) $0 - $100    45.34     1       0      Urban       0
5            6    2) $100 - $200   134.83     0       1  Surburban       0
11           1    3) $200 - $350   211.45     0       1      Urban       1
12           5    5) $500 - $750   642.90     0       1  Surburban       1
13           2    2) $100 - $200   101.64     0       1      Urban       0
14           4    3) $200 - $350   241.42     0       1      Rural       1
15           3      1) $0 - $100    58.13     1       0      Urban       1
16           5      1) $0 - $100    29.99     1       0  Surburban       0
19           5  6) $750 - $1,000   828.42     1       0  Surburban       1
22           2    2) $100 - $200   118.40     1       0  Surburban       0
23           2      1) $0 - $100    29.99     0       1      Urban       1
24           4      1) $0 - $100    78.24     1       0  Surburban       0
25           6    2) $100 - $200   139.87     0       1      Rural       1
27           6    2) $100 - $200   162.98     0       1  Surburban       0
29           2    3) $200 - $350   203.35     1       0      Rural       0
30           2    3) $200 - $350   237.53     0       1  Surburban       0
32           6    2) $100 - $200   128.01     0       1      Urban       0
34           3      1) $0 - $100    29.99     1       0      Rural       0
35           4    3) $200 - $350   218.72     0       1      Urban       0
36           1    5) $500 - $750   514.52     0       1  Surburban       1
37           4  6) $750 - $1,000   766.47     1       1      Urban       1
41           3      1) $0 - $100    99.23     1       0      Rural       0
43           2    4) $350 - $500   492.02     1       0  Surburban       0
44           1      1) $0 - $100    48.32     0       1      Urban       0
46           2    4) $350 - $500   391.33     1       0  Surburban       0
47           1    5) $500 - $750   729.70     1       1  Surburban       1
48           3    2) $100 - $200   134.59     1       0      Urban       1
50           3    3) $200 - $350   203.30     0       1  Surburban       0
55           6      1) $0 - $100    42.66     1       0  Surburban       0
...        ...               ...      ...   ...     ...        ...     ...
63949        5      1) $0 - $100    86.79     1       0      Rural       0
63950        1      1) $0 - $100    45.67     0       1  Surburban       1
63953        5    2) $100 - $200   166.24     0       1      Urban       0
```

63954	2	1) $0 - $100	93.97	1	0	Urban	0
63955	1	1) $0 - $100	29.99	1	0	Surburban	0
63960	1	3) $200 - $350	221.89	0	1	Surburban	1
63961	4	3) $200 - $350	337.36	1	0	Urban	0
63964	2	6) $750 - $1,000	772.99	1	1	Surburban	1
63966	4	2) $100 - $200	170.03	1	0	Surburban	0
63967	5	1) $0 - $100	77.73	0	1	Urban	1
63969	3	1) $0 - $100	67.78	0	1	Surburban	0
63970	4	2) $100 - $200	191.15	0	1	Surburban	1
63971	5	5) $500 - $750	549.87	0	1	Surburban	1
63972	3	5) $500 - $750	554.97	0	1	Surburban	1
63976	1	5) $500 - $750	710.72	1	1	Urban	1
63980	3	4) $350 - $500	487.10	0	1	Surburban	1
63981	4	2) $100 - $200	125.53	0	1	Rural	1
63982	5	1) $0 - $100	29.99	1	0	Urban	1
63983	2	1) $0 - $100	83.03	0	1	Urban	0
63984	2	3) $200 - $350	209.51	0	1	Urban	1
63987	1	1) $0 - $100	79.70	1	0	Surburban	1
63988	6	1) $0 - $100	32.98	1	0	Surburban	0
63990	6	1) $0 - $100	80.02	0	1	Surburban	0
63991	1	3) $200 - $350	306.10	1	0	Surburban	1
63992	1	5) $500 - $750	519.69	1	1	Urban	1
63993	4	4) $350 - $500	374.07	0	1	Surburban	0
63996	5	1) $0 - $100	38.91	0	1	Urban	1
63997	6	1) $0 - $100	29.99	1	0	Urban	1
63998	1	5) $500 - $750	552.94	1	0	Surburban	1
63999	1	4) $350 - $500	472.82	0	1	Surburban	0

	channel	segment	visit	conversion	spend	DM_category
1	Web	No E-Mail	0	0	0.0	11
4	Web	Womens E-Mail	0	0	0.0	4
5	Phone	Womens E-Mail	1	0	0.0	1
11	Phone	Womens E-Mail	0	0	0.0	1
12	Multichannel	Womens E-Mail	0	0	0.0	1
13	Web	Mens E-Mail	1	0	0.0	3
14	Multichannel	No E-Mail	0	0	0.0	5
15	Web	No E-Mail	1	0	0.0	6
16	Phone	Mens E-Mail	0	0	0.0	2
19	Multichannel	Mens E-Mail	0	0	0.0	2
22	Web	Mens E-Mail	1	0	0.0	2
23	Phone	No E-Mail	0	0	0.0	5
24	Web	No E-Mail	0	0	0.0	6
25	Web	Mens E-Mail	0	0	0.0	3
27	Web	Mens E-Mail	0	0	0.0	3
29	Web	No E-Mail	0	0	0.0	6
30	Phone	Womens E-Mail	0	0	0.0	1
32	Web	Mens E-Mail	0	0	0.0	3
34	Web	Womens E-Mail	0	0	0.0	4
35	Multichannel	Womens E-Mail	0	0	0.0	1
36	Web	Mens E-Mail	0	0	0.0	3
37	Multichannel	Mens E-Mail	0	0	0.0	5
41	Web	Mens E-Mail	1	0	0.0	2
43	Phone	No E-Mail	0	0	0.0	6
44	Web	No E-Mail	0	0	0.0	5
46	Web	No E-Mail	0	0	0.0	6
47	Web	Mens E-Mail	0	0	0.0	5
48	Phone	Womens E-Mail	1	0	0.0	4
50	Web	No E-Mail	0	0	0.0	5
55	Web	No E-Mail	0	0	0.0	6
...	...	...	...	...	...	...
63949	Phone	No E-Mail	0	0	0.0	6
63950	Web	Mens E-Mail	0	0	0.0	3
63953	Phone	No E-Mail	0	0	0.0	5
63954	Web	No E-Mail	1	0	0.0	6
63955	Phone	Mens E-Mail	0	0	0.0	2
63960	Multichannel	No E-Mail	0	0	0.0	5
63961	Web	Mens E-Mail	1	0	0.0	2
63964	Web	Mens E-Mail	0	0	0.0	5
63966	Web	Womens E-Mail	0	0	0.0	4
63967	Phone	No E-Mail	0	0	0.0	5
63969	Web	Womens E-Mail	0	0	0.0	1
63970	Web	Womens E-Mail	0	0	0.0	1
63971	Phone	Womens E-Mail	0	0	0.0	1
63972	Web	No E-Mail	0	0	0.0	5
63976	Phone	No E-Mail	0	0	0.0	11
63980	Phone	No E-Mail	0	0	0.0	5
63981	Phone	No E-Mail	0	0	0.0	5
63982	Phone	Mens E-Mail	0	0	0.0	2
63983	Phone	No E-Mail	0	0	0.0	5
63984	Web	Womens E-Mail	0	0	0.0	1

```
63987          Web      No E-Mail      0           0    0.0            6
63988          Web    Mens E-Mail      0           0    0.0            2
63990        Phone      No E-Mail      0           0    0.0            5
63991        Phone  Womens E-Mail      0           0    0.0            4
63992        Phone    Mens E-Mail      0           0    0.0            5
63993        Phone  Womens E-Mail      0           0    0.0            1
63996        Phone    Mens E-Mail      0           0    0.0            3
63997        Phone    Mens E-Mail      0           0    0.0            2
63998  Multichannel  Womens E-Mail      0           0    0.0            4
63999          Web    Mens E-Mail      0           0    0.0            3

[36585 rows x 13 columns]
combine multiple boolean indexing conditions
       recency history_segment  history  mens  womens   zip_code  newbie  \
4            2    1) $0 - $100    45.34     1       0      Urban       0
5            6  2) $100 - $200   134.83     0       1  Surburban       0
13           2  2) $100 - $200   101.64     0       1      Urban       0
16           5    1) $0 - $100    29.99     1       0  Surburban       0
22           2  2) $100 - $200   118.40     1       0  Surburban       0
24           4    1) $0 - $100    78.24     1       0  Surburban       0
27           6  2) $100 - $200   162.98     0       1  Surburban       0
29           2  3) $200 - $350   203.35     1       0      Rural       0
30           2  3) $200 - $350   237.53     0       1  Surburban       0
32           6  2) $100 - $200   128.01     0       1      Urban       0
34           3    1) $0 - $100    29.99     1       0      Rural       0
35           4  3) $200 - $350   218.72     0       1      Urban       0
41           3    1) $0 - $100    99.23     1       0      Rural       0
43           2  4) $350 - $500   492.02     1       0  Surburban       0
44           1    1) $0 - $100    48.32     0       1      Urban       0
46           2  4) $350 - $500   391.33     1       0  Surburban       0
50           3  3) $200 - $350   203.30     0       1  Surburban       0
55           6    1) $0 - $100    42.66     1       0  Surburban       0
59           3  2) $100 - $200   143.93     0       1  Surburban       0
61           2    1) $0 - $100    96.91     1       0  Surburban       0
65           5  3) $200 - $350   222.07     0       1  Surburban       0
70           2  3) $200 - $350   278.80     1       0      Rural       0
72           2  4) $350 - $500   428.74     1       0      Rural       0
76           4  2) $100 - $200   194.11     1       0      Urban       0
79           2    1) $0 - $100    29.99     0       1  Surburban       0
81           2    1) $0 - $100    95.33     0       1  Surburban       0
86           2    1) $0 - $100    82.59     1       0  Surburban       0
87           6  2) $100 - $200   165.77     0       1      Urban       0
95           3  2) $100 - $200   133.51     1       0      Urban       0
97           1    1) $0 - $100    47.44     0       1      Urban       0
...        ...             ...      ...   ...     ...        ...     ...
63890        2  3) $200 - $350   229.79     1       1      Urban       0
63891        4    1) $0 - $100    65.23     0       1      Urban       0
63893        2  3) $200 - $350   288.30     1       1  Surburban       0
63901        4  2) $100 - $200   158.91     1       0  Surburban       0
63902        3  3) $200 - $350   309.39     1       1      Urban       0
63904        1    1) $0 - $100    29.99     0       1  Surburban       0
63906        4  3) $200 - $350   311.63     0       1      Urban       0
63910        6    1) $0 - $100    29.99     1       0  Surburban       0
63911        5  4) $350 - $500   373.13     0       1      Urban       0
63915        4  3) $200 - $350   300.90     1       0  Surburban       0
63916        6    1) $0 - $100    34.50     1       0      Urban       0
63918        3    1) $0 - $100    29.99     1       0  Surburban       0
63919        3  2) $100 - $200   149.96     1       0      Urban       0
63924        4  2) $100 - $200   101.40     0       1      Urban       0
63931        2    1) $0 - $100    29.99     1       0  Surburban       0
63932        1  4) $350 - $500   426.36     1       1  Surburban       0
63935        4  4) $350 - $500   353.47     1       1      Urban       0
63941        2  2) $100 - $200   130.96     1       0      Urban       0
63949        5    1) $0 - $100    86.79     1       0      Rural       0
63953        5  2) $100 - $200   166.24     0       1      Urban       0
63954        2    1) $0 - $100    93.97     1       0      Urban       0
63955        1    1) $0 - $100    29.99     1       0  Surburban       0
63961        4  3) $200 - $350   337.36     1       0      Urban       0
63966        4  2) $100 - $200   170.03     1       0  Surburban       0
63969        3    1) $0 - $100    67.78     0       1  Surburban       0
63983        2    1) $0 - $100    83.03     0       1      Urban       0
63988        6    1) $0 - $100    32.98     1       0  Surburban       0
63990        6    1) $0 - $100    80.02     0       1  Surburban       0
63993        4  4) $350 - $500   374.07     0       1  Surburban       0
63999        1  4) $350 - $500   472.82     0       1  Surburban       0

           channel        segment  visit  conversion  spend  DM_category
4              Web  Womens E-Mail      0           0    0.0            4
5            Phone  Womens E-Mail      1           0    0.0            1
13             Web    Mens E-Mail      1           0    0.0            3
```

```
16           Phone    Mens E-Mail     0          0    0.0          2
22             Web    Mens E-Mail     1          0    0.0          2
24             Web      No E-Mail     0          0    0.0          6
27             Web    Mens E-Mail     0          0    0.0          3
29             Web      No E-Mail     0          0    0.0          6
30           Phone  Womens E-Mail     0          0    0.0          1
32             Web    Mens E-Mail     0          0    0.0          3
34             Web  Womens E-Mail     0          0    0.0          4
35    Multichannel  Womens E-Mail     0          0    0.0          1
41             Web    Mens E-Mail     1          0    0.0          2
43           Phone      No E-Mail     0          0    0.0          6
44             Web      No E-Mail     0          0    0.0          5
46             Web      No E-Mail     0          0    0.0          6
50             Web      No E-Mail     0          0    0.0          5
55             Web      No E-Mail     0          0    0.0          6
59           Phone  Womens E-Mail     0          0    0.0          1
61           Phone  Womens E-Mail     0          0    0.0          4
65           Phone  Womens E-Mail     0          0    0.0          1
70             Web    Mens E-Mail     0          0    0.0          2
72           Phone    Mens E-Mail     0          0    0.0          2
76           Phone      No E-Mail     0          0    0.0          6
79           Phone    Mens E-Mail     0          0    0.0          3
81             Web      No E-Mail     1          0    0.0          5
86           Phone  Womens E-Mail     0          0    0.0          4
87             Web  Womens E-Mail     1          0    0.0          1
95             Web    Mens E-Mail     0          0    0.0          2
97           Phone  Womens E-Mail     0          0    0.0          1
...            ...            ...   ...        ...    ...        ...
63890        Phone    Mens E-Mail     1          0    0.0          5
63891        Phone  Womens E-Mail     0          0    0.0          1
63893        Phone  Womens E-Mail     0          0    0.0          5
63901          Web      No E-Mail     0          0    0.0          6
63902        Phone  Womens E-Mail     0          0    0.0          5
63904          Web  Womens E-Mail     1          0    0.0          1
63906 Multichannel      No E-Mail     0          0    0.0          5
63910        Phone    Mens E-Mail     0          0    0.0          2
63911        Phone  Womens E-Mail     0          0    0.0          1
63915        Phone      No E-Mail     0          0    0.0          6
63916        Phone  Womens E-Mail     0          0    0.0          4
63918          Web      No E-Mail     0          0    0.0          6
63919          Web  Womens E-Mail     0          0    0.0          4
63924          Web      No E-Mail     0          0    0.0          5
63931          Web      No E-Mail     0          0    0.0          6
63932 Multichannel    Mens E-Mail     1          0    0.0          5
63935 Multichannel    Mens E-Mail     0          0    0.0          5
63941        Phone    Mens E-Mail     0          0    0.0          2
63949        Phone      No E-Mail     0          0    0.0          6
63953        Phone      No E-Mail     0          0    0.0          5
63954          Web      No E-Mail     1          0    0.0          6
63955        Phone    Mens E-Mail     0          0    0.0          2
63961          Web    Mens E-Mail     1          0    0.0          2
63966          Web  Womens E-Mail     0          0    0.0          4
63969          Web  Womens E-Mail     0          0    0.0          1
63983        Phone      No E-Mail     0          0    0.0          5
63988          Web    Mens E-Mail     0          0    0.0          2
63990        Phone      No E-Mail     0          0    0.0          5
63993        Phone  Womens E-Mail     0          0    0.0          1
63999          Web    Mens E-Mail     0          0    0.0          3

[17551 rows x 13 columns]
```

```
# df can take in a list of parameters
print(df[['recency']].head(3)) # we will get back a dataframe
print(df.loc[:,['recency']].head(3)) # we will get back a dataframe

print(df.iloc[:, [0]].head(3)) # we will get back a dataframe.
```

```
   recency
0       10
1        6
2        7
   recency
0       10
1        6
2        7
   recency
0       10
1        6
2        7
```

```
# 赋值给切片
df[df.recency<7] = -100
print(df.head())
```

```
   recency history_segment  history  mens  womens   zip_code  newbie channel  \
0       10  2) $100 - $200   142.44     1       0  Surburban       0   Phone
1     -100            -100  -100.00  -100    -100       -100    -100    -100
2        7  2) $100 - $200   180.65     0       1  Surburban       1     Web
3        9  5) $500 - $750   675.83     1       0      Rural       1     Web
4     -100            -100  -100.00  -100    -100       -100    -100    -100

         segment  visit  conversion  spend  DM_category
0  Womens E-Mail      0           0    0.0            4
1           -100   -100        -100 -100.0         -100
2  Womens E-Mail      0           0    0.0            1
3    Mens E-Mail      0           0    0.0            2
4           -100   -100        -100 -100.0         -100
```

6. 编码方法

```
# 简单编码方法
df = pd.DataFrame({
        "哺乳动物":[
            "fish",
            "human",
            "bird",
            "dog"
]})

print("编码前:\n",df)

df.哺乳动物 = df.哺乳动物.astype("category").cat.codes
print("编码后: \n",df)
```

```
编码前:
     哺乳动物
0   fish
1  human
2   bird
3    dog
编码后:
    哺乳动物
0     2
1     3
2     0
3     1
```

```
df = pd.DataFrame({
        "vertebrates":[
            "fish",
            "human",
            "bird",
            "dog"
]})

print(df)

# 注意: 不要忘了中括号 [ ]
```

```
df = pd.get_dummies(df, columns=["vertebrates"])

print(df)
```

```
  vertebrates
0        fish
1       human
2        bird
3         dog
   vertebrates_bird  vertebrates_dog  vertebrates_fish  vertebrates_human
0                 0                0                 1                  0
1                 0                0                 0                  1
2                 1                0                 0                  0
3                 0                1                 0                  0
```

```
# for nominal features: option 1 -- fast and dirty coding method
# 对名义上的特征: 可选方法1: 不那么好但快速的编码方法
df = pd.DataFrame({'vertebrates':[
'Bird',
'Bird',
'Mammal',
'Fish',
'Amphibian',
'Reptile',
'Mammal']})

df['vertebrates'] = df.vertebrates.astype("category").cat.codes
print("不那么好但很快的编码方法:\n",df)

# 可选方法2: 更精确的编码方法
# more accurate coding method

df = pd.get_dummies(df,columns=['vertebrates'])

print('With more accurate method:')
print(df)
```

```
不那么好但很快的编码方法:
    vertebrates
0            1
1            1
2            3
3            2
4            0
5            4
6            3
With more accurate method:
   vertebrates_0  vertebrates_1  vertebrates_2  vertebrates_3  vertebrates_4
0              0              1              0              0              0
1              0              1              0              0              0
2              0              0              0              1              0
3              0              0              1              0              0
4              1              0              0              0              0
5              0              0              0              0              1
6              0              0              0              1              0
```

扩展阅读

Pandas: http://pandas.pydata.org/pandas-docs/stable/cookbook.html

数据处理技术: https://chrisalbon.com/#Python

处理缺失数据: http://pandas.pydata.org/pandas-docs/stable/missing_data.html

Scikit模块家族: https://scikits.appspot.com/scikits

提取特征的技术:--关于词袋模型更多介绍 http://scikit-learn.org/stable/modules/feature_extraction.html#the-bag-of-words-representation

可视化 http://pandas.pydata.org/pandas-docs/stable/visualization.html