Python 数据处理

刘德华

2023 年 6 月 26 日

1 基本技能

1.0.1 以 openml 中的数据集 diabetes 为例，计算不同 class 分类下，age 的中位数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=37,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive

[4]:
```
# 分组计算平均数
data.groupby(by="class")["age"].agg("median")
```

[4]:
```
class
tested_negative    27.0
tested_positive    36.0
Name: age, dtype: float64
```

1.0.2 以 openml 中的数据集 diabetes 为例，查找第五大的年龄是多少岁。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=37,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive

[4]:
```
# 降序排列后取出第四个
data["age"].sort_values(ascending=False).iloc[3]
```

[4]: 69

1.0.3 以 **openml** 中的数据集 **diabetes** 为例，新建一个 **id** 列，划分为两个子数据集，两个数据集的样本量是一样的，只是变量不同。

1.0.4 将这两个的子数据集打乱顺序，按照 **id** 合并。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=37,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive

[4]:
```
# 新建一列 id
data["id"] = range(1,data.shape[0]+1)
```

[5]:
```
# 按照 preg 排序，打乱顺序
data1 = data.loc[:, ["preg", "plas", "pres", "class", "id"]].
 ↪sort_values(by=["preg"])
# 按照 age 排序，打乱顺序
data2 = data.loc[:, ["skin", "insu", "mass", "pedi", "age", "id"]].
 ↪sort_values(by=["age"])
```

[6]:
```
data1.head()
```

[6]:
```
     preg  plas  pres            class   id
467     0    97    64  tested_negative  468
109     0    95    85  tested_positive  110
452     0    91    68  tested_negative  453
449     0   120    74  tested_negative  450
448     0   104    64  tested_positive  449
```

[7]:
```
data2.head()
```

[7]:
```
     skin  insu  mass   pedi  age   id
255    35     0  33.6  0.543   21  256
60      0     0   0.0  0.304   21   61
102     0     0  22.5  0.262   21  103
182    20    23  27.7  0.299   21  183
623    27   115  43.5  0.347   21  624
```

[8]:
```
# 按照 id 合并
newdata = data1.merge(data2, on="id")
# 按照 id 排序
newdata.sort_values(by=["id"], inplace=True)
newdata.head()
```

[8]:
```
     preg  plas  pres            class  id  skin  insu  mass   pedi  age
587     6   148    72  tested_positive   1    35     0  33.6  0.627   50
233     1    85    66  tested_negative   2    29     0  26.6  0.351   31
665     8   183    64  tested_positive   3     0     0  23.3  0.672   32
212     1    89    66  tested_negative   4    23    94  28.1  0.167   21
101     0   137    40  tested_positive   5    35   168  43.1  2.288   33
```

1.0.5 以 openml 中的数据集 1StudentPerfromance 为例，求数学分数 math score 的排名

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
```

```
2  female        group B           master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 按照 max 方法进行排序
res1 = data["math score"].rank(method="max")
res1.sort_values().iloc[:12]
```

[4]:
```
59      1.0
980     2.0
17      3.0
787     4.0
145     5.0
842     6.0
338     7.0
466     8.0
91     10.0
363    10.0
327    11.0
528    14.0
Name: math score, dtype: float64
```

[5]:
```
# 按照 min 方法进行排序
res2 = data["math score"].rank(method="min")
res2.sort_values().iloc[:12]
```

[5]:
```
59      1.0
980     2.0
17      3.0
787     4.0
145     5.0
842     6.0
338     7.0
466     8.0
91      9.0
363     9.0
327    11.0
528    12.0
Name: math score, dtype: float64
```

[6]:
```
# 按照 dense 方法进行排序
res3 = data["math score"].rank(method="dense")
res3.sort_values().iloc[:12]
```

[6]:
```
59      1.0
980     2.0
17      3.0
787     4.0
145     5.0
842     6.0
338     7.0
466     8.0
91      9.0
363     9.0
327    10.0
528    11.0
Name: math score, dtype: float64
```

1.0.6 以 openml 中的数据集 1StudentPerfromance 为例，求数学分数 math score 连续出现 3 次的分数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none			
4	male	group C	some college	standard				

3	none	47	57	44
4	none	76	78	75

[4]:
```
# 一步差分结果，删除缺失值
diff1 = data["math score"].diff().dropna()
diff1
```

[4]:
```
1      -3.0
2      21.0
3     -43.0
4      29.0
5      -5.0
       ...
995    25.0
996   -26.0
997    -3.0
998     9.0
999     9.0
Name: math score, Length: 999, dtype: float64
```

[5]:
```
# 找到为零的那些元素，表示重复两次出现的值
diff1_zero = diff1[diff1==0]
diff1_zero
```

[5]:
```
384    0.0
390    0.0
432    0.0
437    0.0
453    0.0
518    0.0
537    0.0
550    0.0
565    0.0
747    0.0
797    0.0
925    0.0
948    0.0
Name: math score, dtype: float64
```

[6]:
```
# 再对一步差分结果中零的索引进行差分，若为零，则说明是三个连续值
result = pd.Series(diff1_zero.index).diff().dropna()
result[result==0]
```

[6]:
```
Series([], dtype: float64)
```

1.0.7 以 openml 中的数据集 1StudentPerfromance 为例，求阅读分数 reading score 中出现次数大于 1 的重复值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[8]:
```
# 找出 reading score 值相同的行
dup_bool = data["reading score"].duplicated()
res = data["reading score"][dup_bool]
res
```

[8]:
```
6      95
13     72
21     75
22     54
26     54
       ..
995    99
996    55
997    71
998    78
```

```
999    86
Name: reading score, Length: 928, dtype: int64
```

1.0.8 以 openml 中的数据集 1StudentPerfromance 为例，求 reading score 中重复值各自出现的次数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 频数统计
counts = data["reading score"].value_counts()
# 取出频数大于 1 的值
results = counts[counts > 1]
# 按照 reading score 的数值排序
res_sort = results.sort_index(ascending=False)
res_sort
```

[4]:
```
100    17
99      3
97      5
96      4
```

```
95      8
       ..
37      3
34      4
31      2
29      2
24      2
Name: reading score, Length: 66, dtype: int64
```

1.0.9 以 openml 中的数据集 1StudentPerfromance 为例，求 math score 中分数为零的样本。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 数学分数为零的样本
data[data["math score"] == 0]
```

[4]:
```
    gender race/ethnicity parental level of education         lunch  \
59  female        group C            some high school  free/reduced
```

	test preparation course	math score	reading score	writing score
59	none	0	17	10

1.0.10 以 openml 中的数据集 1StudentPerfromance 为例，删除 writing score 中分数相同的样本。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
newdata = data.drop_duplicates(subset=["writing score"])
print("删除 writing score 重复值后的样本量: ", newdata.shape, sep="\n")
newdata.head()
```

```
删除 writing score 重复值后的样本量:
(77, 8)
```

[4]:

	gender	race/ethnicity	parental level of education	lunch
0	female	group B	bachelor\'s degree	standard
1	female	group C	some college	standard
2	female	group B	master\'s degree	standard
3	male	group A	associate\'s degree	free/reduced

```
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

1.0.11 以 openml 中的数据集 1StudentPerfromance 为例，找到下一个 math score 分数比上一个高的样本值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 一步差分结果
diff_res = data["math score"].diff().dropna()
# 从一步差分结果的索引找到这些样本
data.loc[diff_res[diff_res > 0].index, ].head()
```

[4]:
```
    gender race/ethnicity parental level of education         lunch  \
2   female        group B            master\'s degree      standard
4     male        group C                some college      standard
6   female        group B                some college      standard
8     male        group D                 high school  free/reduced
10    male        group C          associate\'s degree      standard

   test preparation course  math score  reading score  writing score
2                     none          90             95             93
4                     none          76             78             75
6                completed          88             95             92
8                completed          64             64             67
10                    none          58             54             52
```

1.0.12 以 openml 中的数据集 1StudentPerfromance 为例，找到 writing score 中分数大于 90 的样本。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B           bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 分数大于 90 的样本
data[data["writing score"] > 90].head()
```

[4]:
```
     gender race/ethnicity parental level of education         lunch  \
2    female        group B            master\'s degree      standard
6    female        group B                some college      standard
94   female        group B                some college      standard
106  female        group D            master\'s degree      standard
110  female        group D         associate\'s degree  free/reduced

    test preparation course  math score  reading score  writing score
2                      none          90             95             93
6                 completed          88             95             92
94                     none          79             86             92
106                    none          87            100            100
110               completed          77             89             98
```

1.0.13 以 openml 中的数据集 1StudentPerfromance 为例，统计各个民族的人数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
```

```
4                   none          76             78             75
```

[4]:
```
# 分组统计频数
data["race/ethnicity"].value_counts()
```

[4]:
```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: race/ethnicity, dtype: int64
```

1.0.14 以 openml 中的数据集 1StudentPerfromance 为例，寻找 math score 中只出现一次的最大数值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 寻找唯一值，再从唯一值中求最大值
data["math score"].unique().max()
```

[4]:
```
100
```

1.0.15 以 **openml** 中的数据集 **1StudentPerfromance** 为例，将 **reading score** 中数值为 **59** 的全部修改为 **60**。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 将 reading score 为 59 的替换为 60
print("替换之前，数字 60 的个数为：", data["reading score"].value_counts().loc[60,])
data["reading score"].replace(to_replace=59, value=60, inplace=True)
print("替换之后，数字 60 的个数为：", data["reading score"].value_counts().loc[60,])
```

```
替换之前，数字 60 的个数为： 21
替换之后，数字 60 的个数为： 38
```

1.0.16 以 **openml** 中的数据集 **CSM** 为例，计算评分 **Ratings** 前 **10** 的电影到底有多少人喜欢？

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=42371,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	Sentiment	\
0	2014	6.3	8	9130	4000000.0	45.0	1	0	
1	2014	7.1	1	192000000	50000000.0	3306.0	2	2	
2	2014	6.2	1	30700000	28000000.0	2872.0	1	0	
3	2014	6.3	1	106000000	110000000.0	3470.0	2	0	
4	2014	4.7	8	17300000	3500000.0	2310.0	2	0	

	Views	Likes	Dislikes	Comments	Aggregate.Followers
0	3280543	4632	425	636	1120000.0
1	583289	3465	61	186	12350000.0
2	304861	328	34	47	483000.0
3	452917	2429	132	590	568000.0
4	3145573	12163	610	1082	1923800.0

[4]:
```
# 按照 Rating 降序排列，取出前 10 对应的 Likes 数
data.sort_values(by=["Ratings"], ascending=False).iloc[:10, ][["Ratings", "Likes"]]
```

[4]:

	Ratings	Likes
55	8.7	16635
166	8.6	4632
155	8.6	17541
174	8.3	13030
175	8.3	12607
158	8.2	1023
45	8.2	1390
212	8.2	18398
46	8.1	8567
129	8.1	11748

1.0.17 以 openml 中的数据集 CSM 为例，将数据集按照评分 Ratings 作为第一因子降序排列，Likes 作为第二因子升序排列。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=42371,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	Sentiment	\
0	2014	6.3	8	9130	4000000.0	45.0	1	0	
1	2014	7.1	1	192000000	50000000.0	3306.0	2	2	
2	2014	6.2	1	30700000	28000000.0	2872.0	1	0	
3	2014	6.3	1	106000000	110000000.0	3470.0	2	0	
4	2014	4.7	8	17300000	3500000.0	2310.0	2	0	

	Views	Likes	Dislikes	Comments	Aggregate.Followers
0	3280543	4632	425	636	1120000.0
1	583289	3465	61	186	12350000.0
2	304861	328	34	47	483000.0
3	452917	2429	132	590	568000.0
4	3145573	12163	610	1082	1923800.0

[4]:
```
# 多字段排序，升降序排列
res = data.sort_values(by=["Ratings", "Likes"], ascending=[False, True])
res.head()
```

[4]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	Sentiment	\
55	2014	8.7	2	188000000	165000000.0	3561.0	1	2	
166	2015	8.6	12	345000000	175000000.0	3946.0	1	2	
155	2014	8.6	3	13100000	3300000.0	42.0	1	2	
175	2015	8.3	9	135000000	28000000.0	2757.0	1	5	
174	2015	8.3	1	153000000	150000000.0	3702.0	4	-4	

	Views	Likes	Dislikes	Comments	Aggregate.Followers
55	5421705	16635	751	4316	1865000.0
166	1438926	4632	262	496	232000.0
155	7750223	17541	631	2760	858000.0
175	848970	12607	237	1560	55618.0
174	2732371	13030	497	1774	768700.0

1.0.18 以 **openml** 中的数据集 **CSM** 为例，将数据集的行索引修改为 **Movie{i}**。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=42371,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	Sentiment	\
0	2014	6.3	8	9130	4000000.0	45.0	1	0	
1	2014	7.1	1	192000000	50000000.0	3306.0	2	2	
2	2014	6.2	1	30700000	28000000.0	2872.0	1	0	
3	2014	6.3	1	106000000	110000000.0	3470.0	2	0	
4	2014	4.7	8	17300000	3500000.0	2310.0	2	0	

	Views	Likes	Dislikes	Comments	Aggregate.Followers
0	3280543	4632	425	636	1120000.0
1	583289	3465	61	186	12350000.0
2	304861	328	34	47	483000.0
3	452917	2429	132	590	568000.0
4	3145573	12163	610	1082	1923800.0

[4]:
```
# 重新设置行索引
data.index = ["Movie{}".format(i) for i in range(1, data.shape[0]+1)]
data.head()
```

[4]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	\
Movie1	2014	6.3	8	9130	4000000.0	45.0	1	
Movie2	2014	7.1	1	192000000	50000000.0	3306.0	2	
Movie3	2014	6.2	1	30700000	28000000.0	2872.0	1	
Movie4	2014	6.3	1	106000000	110000000.0	3470.0	2	
Movie5	2014	4.7	8	17300000	3500000.0	2310.0	2	

	Sentiment	Views	Likes	Dislikes	Comments	Aggregate.Followers
Movie1	0	3280543	4632	425	636	1120000.0
Movie2	2	583289	3465	61	186	12350000.0
Movie3	0	304861	328	34	47	483000.0
Movie4	0	452917	2429	132	590	568000.0

```
Movie5          0  3145573  12163       610      1082            1923800.0
```

1.0.19 以 **openml** 中的数据集 **CSM** 为例，将数据集的部分列名进行修改。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=42371,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	Year	Ratings	Genre	Gross	Budget	Screens	Sequel	Sentiment	Views	Likes	Dislikes	Comments	Aggregate.Followers
0	2014	6.3	8	9130	4000000.0	45.0	1	0	3280543	4632	425	636	1120000.0
1	2014	7.1	1	192000000	50000000.0	3306.0	2	2	583289	3465	61	186	12350000.0
2	2014	6.2	1	30700000	28000000.0	2872.0	1	0	304861	328	34	47	483000.0
3	2014	6.3	1	106000000	110000000.0	3470.0	2	0	452917	2429	132	590	568000.0
4	2014	4.7	8	17300000	3500000.0	2310.0	2	0	3145573	12163	610	1082	1923800.0

[4]:
```
# 列重新命名
data.columns = [
    "年份",
    "评分",
    "类型",
    "gross",
    "预算",
    "screens",
    "续集",
    "感情",
    "观看次数",
    "喜欢人数",
    "不喜欢人数",
```

```
    "评论数",
    "粉丝数"
]
data.head()
```

[4]:
```
    年份   评分  类型      gross           预算  screens  续集  感情     观看次数  ␣
↪喜欢人数  \
0  2014  6.3   8       9130    4000000.0     45.0   1   0  3280543   4632
1  2014  7.1   1  192000000   50000000.0   3306.0   2   2   583289   3465
2  2014  6.2   1   30700000   28000000.0   2872.0   1   0   304861    328
3  2014  6.3   1  106000000  110000000.0   3470.0   2   0   452917   2429
4  2014  4.7   8   17300000    3500000.0   2310.0   2   0  3145573  12163

   不喜欢人数   评论数         粉丝数
0    425   636   1120000.0
1     61   186  12350000.0
2     34    47    483000.0
3    132   590    568000.0
4    610  1082   1923800.0
```

1.0.20 以 **openml** 中的数据集 **CSM** 为例，从数据集中无放回地随机抽取 **50** 个样本，计算评分 **Ratings** 的平均值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=42371,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   Year  Ratings  Genre      Gross       Budget  Screens  Sequel  Sentiment  \
0  2014      6.3      8       9130    4000000.0     45.0       1          0
1  2014      7.1      1  192000000   50000000.0   3306.0       2          2
2  2014      6.2      1   30700000   28000000.0   2872.0       1          0
3  2014      6.3      1  106000000  110000000.0   3470.0       2          0
4  2014      4.7      8   17300000    3500000.0   2310.0       2          0

     Views  Likes  Dislikes  Comments  Aggregate.Followers
```

```
0  3280543   4632       425       636            1120000.0
1   583289   3465        61       186           12350000.0
2   304861    328        34        47             483000.0
3   452917   2429       132       590             568000.0
4  3145573  12163       610      1082            1923800.0
```

[4]:
```
# 随机抽样平均值
data["Ratings"].sample(n=50, replace=False).mean()
```

[4]:
```
6.534
```

1.0.21 以 openml 中的数据集 1StudentPerformance 为例，计算 math score 大于 60 且小于 90 的人数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 查找满足条件的样本
res = data[(data["math score"] > 60) & (data["math score"] < 90)]
# 计算样本量
res.shape[0]
```

[4]: 603

1.0.22 以 openml 中的数据集 1StudentPerformance 为例，找到 math score 小于 reading score 小于 writing score 的人。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 修改列名，为了使用 query 语法
data.columns = data.columns[:5].to_list() + ["math_score", "reading_score",␣
↪"writing_score"]
# 该查询方法直接是列名表达式
newdata = data.query("math_score < reading_score < writing_score")
newdata.head()
```

[4]:

	gender	race/ethnicity	parental level of education	lunch	\
14	female	group A	master\'s degree	standard	
15	female	group C	some high school	standard	
19	female	group C	associate\'s degree	free/reduced	
27	female	group C	bachelor\'s degree	standard	
29	female	group D	master\'s degree	standard	

	test preparation course	math_score	reading_score	writing_score
14	none	50	53	58
15	none	69	75	78
19	none	54	58	61
27	none	67	69	75
29	none	62	70	75

1.0.23 以 openml 中的数据集 1StudentPerformance 为例，查找 math score 中是否存在零分。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	\
0	female	group B	bachelor\'s degree	standard	
1	female	group C	some college	standard	
2	female	group B	master\'s degree	standard	
3	male	group A	associate\'s degree	free/reduced	
4	male	group C	some college	standard	

	test preparation course	math score	reading score	writing score
0	none	72	72	74
1	completed	69	90	88
2	none	90	95	93
3	none	47	57	44
4	none	76	78	75

[4]:
```
0 in data["math score"]
```

[4]: True

1.0.24 以 **openml** 中的数据集 **1StudentPerformance** 为例，查找 **math score** 的数值属于 **reading score** 的样本。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 修改列名，为了使用 query 语法
data.columns = data.columns[:5].to_list() + ["math", "reading", "writing"]
res = data.query("math in reading")
res.head()
```

[4]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math  reading  writing
0                    none    72       72       74
1               completed    69       90       88
```

```
2                    none     90        95        93
3                    none     47        57        44
4                    none     76        78        75
```

1.0.25 以 **openml** 中的数据集 **1StudentPerformance** 为例，查找 **race/ethnicity** 为 **groupA** 和 **groupB** 的人。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 修改列名
data["race"] = data["race/ethnicity"].copy()
# 删除原来的列
data.drop(columns=["race/ethnicity"], inplace=True)
res = data.query("race in ['group A', 'group C']")
res.head()
```

[4]:
```
   gender parental level of education         lunch test preparation course  \
1  female                some college      standard                completed
3    male         associate\'s degree  free/reduced                     none
```

```
4     male                some college      standard                    none
10    male         associate\'s degree      standard                    none
13    male                some college      standard               completed

    math score  reading score  writing score     race
1           69             90             88  group C
3           47             57             44  group A
4           76             78             75  group C
10          58             54             52  group C
13          78             72             70  group A
```

1.0.26 以 openml 中的数据集 1StudentPerformance 为例，将列名 math score 修改为 MathScore，将 reading score 修改为 ReadingScore，将 writing score 修改为 WritingScore。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[ ]:
```
# 查看数据集的前五行
data.head()
```

[ ]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[ ]:
```
# 修改列名
data.rename(columns={
    "math score": "MathScore",
```

```
    "reading score": "ReadingScore",
    "writing score": "WritingScore",
}, inplace=True)
data.head()
```

1.0.27 以 openml 中的数据集 1StudentPerformance 为例，将数据集随机地拆分为三块，再从三个子集中分别进行重抽样，将所得到的结果按行合并。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 数据集划分
from sklearn.model_selection import train_test_split
# 第一次划分
df_train, df_test = train_test_split(data, test_size=0.3, random_state=1)
# 第二次划分
df_train1, df_test1 = train_test_split(df_train, test_size=0.3, random_state=2)
# 重抽样
sample1 = df_train1.sample(n=100, replace=True)
sample2 = df_test1.sample(n=100, replace=True)
```

```
sample3 = df_test.sample(n=100, replace=True)
# 按行合并
newdata = pd.concat([sample1, sample2, sample3], axis=0)
newdata.head()
```

[4]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
497	female	group D	some college	free/reduced	completed	59	78	76
384	female	group A	some high school	free/reduced	none	38	43	43
487	female	group C	associate\'s degree	free/reduced	none	60	75	74
273	female	group D	some college	standard	none	65	70	71
728	female	group D	high school	free/reduced	none	73	92	84

1.0.28 以 **openml** 中的数据集 **1StudentPerformance** 为例，将数据集进行三次重抽样，将所得到的结果按行合并（每次重抽样的结果可能有样本值是相同的，合并时需要特别注意，我们这里重新设置行索引）。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard				
1	female	group C	some college	standard				
2	female	group B	master\'s degree	standard				
3	male	group A	associate\'s degree	free/reduced				
4	male	group C	some college	standard				

```
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[8]:
```
# 重抽样
sample1 = data.sample(n=100, replace=True, random_state=1)
sample2 = data.sample(n=100, replace=True, random_state=2)
sample3 = data.sample(n=100, replace=True, random_state=3)
# 按行合并，忽略行索引，重新给定索引
newdata = pd.concat([sample1, sample2, sample3], axis=0, ignore_index=True)
newdata.head()
```

[8]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group D            some high school  free/reduced
1    male        group D         associate\'s degree      standard
2  female        group C          bachelor\'s degree  free/reduced
3  female        group A         associate\'s degree  free/reduced
4    male        group B                 high school      standard

  test preparation course  math score  reading score  writing score
0                    none          50             64             59
1                    none          80             75             77
2                    none          67             75             72
3                    none          41             51             48
4               completed          76             62             60
```

1.0.29 以 **openml** 中的数据集 **1StudentPerformance** 为例，将数据集拆分成两个子集（变量拆分），样本是从原数据集中重抽样所得，将这两个子集按列合并，默认键是行索引 **index**。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 重抽样
data1 = data.iloc[:, :4].sample(n=50, replace=True, random_state=1)
data2 = data.iloc[:, 4:].sample(n=50, replace=True, random_state=2)
# 按列合并，默认的 key 是 Index，取并集，以缺失值填充
newdata1 = pd.concat([data1, data2], axis=1, join="outer")
newdata1.head()
```

[4]:
```
     gender race/ethnicity parental level of education         lunch  \
37   female        group D            some high school  free/reduced
235    male        group D          associate\'s degree      standard
908  female        group C           bachelor\'s degree  free/reduced
72   female        group A          associate\'s degree  free/reduced
767    male        group B                 high school      standard

    test preparation course  math score  reading score  writing score
37                      NaN         NaN            NaN            NaN
235                     NaN         NaN            NaN            NaN
908                     NaN         NaN            NaN            NaN
72                      NaN         NaN            NaN            NaN
767                     NaN         NaN            NaN            NaN
```

[5]:
```
# 按列合并，默认的 key 是 Index，取交集，无缺失值
newdata2 = pd.concat([data1, data2], axis=1, join="inner")
newdata2.head()
```

[5]:
```
    gender race/ethnicity parental level of education     lunch  \
534   male        group B                high school  standard

    test preparation course  math score  reading score  writing score
534               completed          73             69             68
```

1.0.30 以 openml 中的数据集 1StudentPerformance 为例，将列 gender 取出来转为 dataframe 对象。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# Series 转 Dataframe
data["gender"].to_frame()
```

[4]:
```
      gender
0     female
1     female
2     female
3       male
4       male
..       ...
995   female
996     male
997   female
998   female
999   female
```

```
[1000 rows x 1 columns]
```

1.0.31 以 openml 中的数据集 1StudentPerformance 为例，统计不同 race 和不同性别下数学成绩的平均值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 在不同的分组下，计算数学成绩平均值
data.groupby(by=["gender", "race/ethnicity"])["math score"].agg("mean")
```

[4]:
```
gender  race/ethnicity
female  group A           58.527778
        group B           61.403846
        group C           62.033333
        group D           65.248062
        group E           70.811594
male    group A           63.735849
        group B           65.930233
        group C           67.611511
```

```
        group D           69.413534
        group E           76.746479
Name: math score, dtype: float64
```

1.0.32 以 **openml** 中的数据集 **1StudentPerformance** 为例，统计不同 **race** 和不同性别下阅读成绩的中位数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 数据透视表
pd.pivot_table(
    data,
    values="reading score",
    index="gender",
    columns="race/ethnicity",
    aggfunc="median"
)
```

[4]:
```
race/ethnicity  group A  group B  group C  group D  group E
gender
female             67.5     71.5     73.0     74.0     76.0
male               61.0     62.0     66.0     68.0     73.0
```

1.0.33 以 openml 中的数据集 1StudentPerformance 为例，统计不同 race 和不同性别，不同的 parental level of education 下，写作成绩的方差。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 数据透视表
pd.pivot_table(
    data,
    values="writing score",
    index=["gender", "parental level of education"],
    columns="race/ethnicity",
    aggfunc="var"
)
```

[4]:
```
race/ethnicity                        group A     group B     group C  \
gender parental level of education
female associate\'s degree          260.966667  152.221344  144.846465
       bachelor\'s degree            94.333333   59.290909  158.426154
       high school                  219.476190  231.485450  210.171264
       master\'s degree             112.500000  171.200000  157.571429
       some college                 167.696429  236.352381  220.390592
       some high school             358.711111  284.874459  314.109788
male   associate\'s degree          244.125000  217.911765  211.751894
       bachelor\'s degree           161.527778  180.777778  223.170330
       high school                  120.654545  234.892105  129.466132
       master\'s degree                    NaN         NaN  126.060606
       some college                 396.722222  233.922078  262.640000
       some high school             150.131868  186.495833   85.447619

race/ethnicity                        group D     group E
gender parental level of education
female associate\'s degree          215.449275  192.970588
       bachelor\'s degree           214.858974  387.333333
       high school                  181.441176  207.659091
       master\'s degree             228.780952  306.666667
       some college                 110.063866  152.229167
       some high school             232.090000  387.766667
male   associate\'s degree          149.435385  226.147619
       bachelor\'s degree           203.095238  141.267857
       high school                  158.396011  109.833333
       master\'s degree              47.071429         NaN
       some college                 163.741935  144.368421
       some high school             173.156667  276.931818
```

1.0.34 以 openml 中的数据集 1StudentPerformance 为例，统计不同 race 和不同性别，不同的 parental level of education 下，样本的数量，并给出边际值（也是数量）。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 数据透视表
pd.pivot_table(
    data,
    values="writing score",
    index=["gender", "parental level of education"],
    columns="race/ethnicity",
    aggfunc="count",
    margins=True,
    margins_name="样本总数"
)
```

[4]:
```
race/ethnicity                      group A  group B  group C  group D  \
gender parental level of education
female associate\'s degree                6       23       45       24
       bachelor\'s degree                 3       11       26       13
       high school                        7       28       30       17
       master\'s degree                   2        5        7       15
       some college                       8       15       44       35
       some high school                  10       22       28       25
male   associate\'s degree                8       18       33       26
       bachelor\'s degree                 9        9       14       15
       high school                       11       20       34       27
       master\'s degree                   1        1       12        8
       some college                      10       22       25       32
       some high school                  14       16       21       25
样本总数                                  89      190      319      262

race/ethnicity                      group E  样本总数
gender parental level of education
```

```
female associate\'s degree              18   116
       bachelor\'s degree               10    63
       high school                      12    94
       master\'s degree                  7    36
       some college                     16   118
       some high school                  6    91
male   associate\'s degree              21   106
       bachelor\'s degree                8    55
       high school                      10   102
       master\'s degree                  1    23
       some college                     19   108
       some high school                 12    88
样本总数                                   140  1000
```

1.0.35 以 **openml** 中的数据集 **1StudentPerformance** 为例，计算 **gender** 和 **race/ethnicity** 的二维列联表。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 列联表
pd.crosstab(
    index=data["gender"],
    columns=data["race/ethnicity"]
)
```

[4]:
```
race/ethnicity  group A  group B  group C  group D  group E
gender
female               36      104      180      129       69
male                 53       86      139      133       71
```

1.0.36 以 **openml** 中的数据集 **1StudentPerformance** 为例，计算 **gender** 和 **race/ethnicity** 和 **lunch** 的三维频率列联表，并给出边际值。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 列联表
pd.crosstab(
    index=[data["gender"], data["lunch"]], # 必须是列表形式
    columns=data["race/ethnicity"],
```

```
    normalize=True,
    margins=True,
    margins_name="合计比例"
)
```

[4]:
```
race/ethnicity       group A  group B  group C  group D  group E   合计比例
gender lunch
female free/reduced    0.014    0.039    0.062    0.051    0.023  0.189
       standard        0.022    0.065    0.118    0.078    0.046  0.329
male   free/reduced    0.022    0.030    0.052    0.044    0.018  0.166
       standard        0.031    0.056    0.087    0.089    0.053  0.316
合计比例                   0.089    0.190    0.319    0.262    0.140  1.000
```

1.0.37 以 **openml** 中的数据集 **1StudentPerformance** 为例，对数学成绩 **math score** 离散化分组。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 离散变量分组
pd.cut(data["math score"], bins=10)
```

[4]:
```
0      (70.0, 80.0]
1      (60.0, 70.0]
2      (80.0, 90.0]
3      (40.0, 50.0]
4      (70.0, 80.0]
          …
995    (80.0, 90.0]
996    (60.0, 70.0]
997    (50.0, 60.0]
998    (60.0, 70.0]
999    (70.0, 80.0]
Name: math score, Length: 1000, dtype: category
Categories (10, interval[float64, right]): [(-0.1, 10.0] < (10.0, 20.0] < (20.0,
30.0] < (30.0, 40.0] … (60.0, 70.0] < (70.0, 80.0] < (80.0, 90.0] < (90.0,
100.0]]
```

1.0.38 以 **openml** 中的数据集 **1StudentPerformance** 为例，对 **gender** 进行虚拟变量化。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 生成虚拟变量
dummy = pd.get_dummies(data["gender"], prefix="性别")
dummy
```

[4]:
```
     性别 _female  性别 _male
0             1         0
1             1         0
2             1         0
3             0         1
4             0         1
..          …       …
995           1         0
996           0         1
997           1         0
998           1         0
999           1         0

[1000 rows x 2 columns]
```

[5]:
```
# 从虚拟变量的 dataframe 转为正常的一列 dataframe
newdata = pd.from_dummies(dummy)
newdata
```

[5]:
```
0    性别 _female
1    性别 _female
2    性别 _female
3      性别 _male
4      性别 _male
..          …
995  性别 _female
996    性别 _male
997  性别 _female
998  性别 _female
999  性别 _female

[1000 rows x 1 columns]
```

1.0.39 以 openml 中的数据集 1StudentPerformance 为例，对 gender 转为因子类型。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 变量因子化
newdata, index = pd.factorize(data["gender"])
newdata[:10]
```

[4]:
```
array([0, 0, 0, 1, 1, 0, 0, 1, 1, 0], dtype=int64)
```

[5]:
```
index
```

[5]:
```
CategoricalIndex(['female', 'male'], categories=['female', 'male'],
ordered=False, dtype='category')
```

1.0.40 以 openml 中的数据集 1StudentPerformance 为例，将宽数据转为长数据。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
```

```
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 宽数据转为长数据
data.iloc[:5, [0,1,5,6]].melt(id_vars=["gender", "race/ethnicity"])
```

[4]:
```
   gender race/ethnicity       variable  value
0  female        group B     math score     72
1  female        group C     math score     69
2  female        group B     math score     90
3    male        group A     math score     47
4    male        group C     math score     76
5  female        group B  reading score     72
6  female        group C  reading score     90
7  female        group B  reading score     95
8    male        group A  reading score     57
9    male        group C  reading score     78
```

1.0.41 以 openml 中的数据集 1StudentPerformance 为例，将 gender 的字符串全部大写，将 race/ethnicity 的字符串全部小写。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
```

```
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 字符串大小写
data["gender"].str.upper()
```

[4]:
```
0      FEMALE
1      FEMALE
2      FEMALE
3        MALE
4        MALE
        ...
995    FEMALE
996      MALE
997    FEMALE
998    FEMALE
999    FEMALE
Name: gender, Length: 1000, dtype: object
```

[5]:
```
data["race/ethnicity"].str.lower()
```

[5]:
```
0      group b
1      group c
2      group b
3      group a
4      group c
        ...
995    group e
996    group c
997    group c
```

```
998    group d
999    group d
Name: race/ethnicity, Length: 1000, dtype: object
```

1.0.42 以 openml 中的数据集 1StudentPerformance 为例，计算 lunch 列字符串中字符的个数。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 字符个数
data["lunch"].str.len()
```

[4]:
```
0       8
1       8
2       8
3      12
4       8
       ..
995     8
996    12
```

```
997    12
998     8
999    12
Name: lunch, Length: 1000, dtype: int64
```

1.0.43 以 openml 中的数据集 1StudentPerformance 为例，将 race/ethnicity 中的字符串空格去除。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 去除空格
data["race/ethnicity"].str.replace(" ", "")
```

[4]:
```
0      groupB
1      groupC
2      groupB
3      groupA
4      groupC
        ...
995    groupE
```

```
996    groupC
997    groupC
998    groupD
999    groupD
Name: race/ethnicity, Length: 1000, dtype: object
```

1.0.44 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **race/ethnicity** 的字符串按照空格分割成列表。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 分割为列表
strlist = data["race/ethnicity"].str.split(" ")
strlist
```

[4]:
```
0      [group, B]
1      [group, C]
2      [group, B]
3      [group, A]
```

```
4      [group, C]
          …
995    [group, E]
996    [group, C]
997    [group, C]
998    [group, D]
999    [group, D]
Name: race/ethnicity, Length: 1000, dtype: object
```

[5]:
```
# 获取分割的列表的第二个元素
strlist.str[1]
```

[5]:
```
0      B
1      C
2      B
3      A
4      C
      ..
995    E
996    C
997    C
998    D
999    D
Name: race/ethnicity, Length: 1000, dtype: object
```

1.0.45 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **race/ethnicity** 的字符串按照空格分割成列表，并将列表中的各个元素作为列加入到 **dataframe** 中。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
```

```
2  female        group B           master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 分割为列表
strlist = data["race/ethnicity"].str.split(" ", expand=True)
strlist
```

[4]:
```
         0  1
0    group  B
1    group  C
2    group  B
3    group  A
4    group  C
..     … ..
995  group  E
996  group  C
997  group  C
998  group  D
999  group  D

[1000 rows x 2 columns]
```

1.0.46 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **parental level of education** 的字符串中的反斜杠去掉。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 去掉反斜杠，使用替换
data["parental level of education"].iloc[0]
```

[4]:
```
"bachelor\\'s degree"
```

[5]:
```
# 使用正则表达式替换
data["parental level of education"].str.replace("\\", "", regex=True)
```

[5]:
```
0       bachelor's degree
1            some college
2         master's degree
3      associate's degree
4            some college
              ...
995       master's degree
996           high school
997           high school
998          some college
999          some college
Name: parental level of education, Length: 1000, dtype: object
```

1.0.47 以 openml 中的数据集 1StudentPerformance 为例，将 math score，reading score 和 writing score 三列合并为一列，以逗号分隔。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor\'s degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master\'s degree	standard	none	90	95	93
3	male	group A	associate\'s degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[4]:
```
# 列之间的合并
newS = data["math score"].astype("string").str.cat(data["reading score"].
↪astype("string"), sep=",").str.cat(data["writing score"].astype("string"),␣
↪sep=",")
newS
```

[4]:
```
0      72,72,74
1      69,90,88
2      90,95,93
3      47,57,44
4      76,78,75
         …
995    88,99,95
996    62,55,55
997    59,71,65
998    68,78,77
999    77,86,86
Name: math score, Length: 1000, dtype: string
```

1.0.48 对含有缺失值的列进行求和和求乘积，看看结果如何

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

[2]:
```
# 生成 dataframe
df = pd.DataFrame(
    np.random.randn(5, 3),
    index=["a", "c", "e", "f", "h"],
    columns=["one", "two", "three"],
)
df["four"] = "bar"
df["five"] = df["one"] > 0
df = df.reindex(["a", "b", "c", "d", "e", "f", "g", "h"])
df
```

[2]:

	one	two	three	four	five
a	0.271842	-0.350975	0.194960	bar	True
b	NaN	NaN	NaN	NaN	NaN
c	-0.104840	-0.469391	-0.410844	bar	False
d	NaN	NaN	NaN	NaN	NaN
e	-0.059557	0.436472	-0.792940	bar	False
f	0.575772	-0.903377	0.192159	bar	True
g	NaN	NaN	NaN	NaN	NaN
h	-1.555696	0.607744	-0.759766	bar	False

[3]:
```
# 计算含有缺失值的列的和，默认是跳过缺失值
df["one"].sum()
```

[3]:
```
-0.8724777568039623
```

[4]:
```
# 不跳过缺失值，那么得到的和就是 NaN
df["two"].sum(skipna=False)
```

[4]:
```
nan
```

[5]:
```
# 计算含有缺失值的列的积，默认是跳过缺失值
df["one"].prod()
```

[5]:
```
-0.0015203634461877212
```

[6]:
```
# 不跳过缺失值，那么得到的积就是 NaN
df["two"].prod(skipna=False)
```

[6]:
```
nan
```

1.0.49 对含有缺失值的列进行填充，将缺失值以其他的值替换

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

[2]:
```
# 生成 dataframe
df = pd.DataFrame(
    np.random.randn(5, 3),
    index=["a", "c", "e", "f", "h"],
    columns=["one", "two", "three"],
)
df["four"] = "bar"
df["five"] = df["one"] > 0
df = df.reindex(["a", "b", "c", "d", "e", "f", "g", "h"])
df
```

[2]:

	one	two	three	four	five
a	1.010471	1.775946	0.116074	bar	True
b	NaN	NaN	NaN	NaN	NaN
c	-0.709978	0.945866	1.369733	bar	False
d	NaN	NaN	NaN	NaN	NaN
e	0.010790	0.209315	1.177950	bar	True
f	0.028884	1.194446	-0.041722	bar	True
g	NaN	NaN	NaN	NaN	NaN
h	1.731725	0.499953	-0.015604	bar	True

[3]:
```
# 以均值替换
df["one"].fillna(df["one"].mean())
```

[3]:
```
a    1.010471
b    0.414378
c   -0.709978
d    0.414378
e    0.010790
f    0.028884
g    0.414378
h    1.731725
Name: one, dtype: float64
```

1.0.50 对含有缺失值的列或者行删除。

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

[2]:
```
# 生成 dataframe
df = pd.DataFrame(
    np.random.randn(5, 3),
```

```
    index=["a", "c", "e", "f", "h"],
    columns=["one", "two", "three"],
)
df["four"] = "bar"
df["five"] = df["one"] > 0
df = df.reindex(["a", "b", "c", "d", "e", "f", "g", "h"])
df
```

[2]:

	one	two	three	four	five
a	1.105184	0.696500	1.024294	bar	True
b	NaN	NaN	NaN	NaN	NaN
c	0.286259	-1.286451	1.746801	bar	True
d	NaN	NaN	NaN	NaN	NaN
e	0.283374	-0.699082	1.299307	bar	True
f	-0.698294	0.484663	1.120755	bar	False
g	NaN	NaN	NaN	NaN	NaN
h	-2.119560	0.173416	0.700130	bar	False

[3]:
```
# 删除含有缺失值的行，存在缺失值就删除
df.dropna(axis=0, how="any")
```

[3]:

	one	two	three	four	five
a	1.105184	0.696500	1.024294	bar	True
c	0.286259	-1.286451	1.746801	bar	True
e	0.283374	-0.699082	1.299307	bar	True
f	-0.698294	0.484663	1.120755	bar	False
h	-2.119560	0.173416	0.700130	bar	False

[4]:
```
# 删除含有缺失值的列，全部都是缺失值才删除
df.dropna(axis=1, how="all")
```

[4]:

	one	two	three	four	five
a	1.105184	0.696500	1.024294	bar	True
b	NaN	NaN	NaN	NaN	NaN
c	0.286259	-1.286451	1.746801	bar	True
d	NaN	NaN	NaN	NaN	NaN
e	0.283374	-0.699082	1.299307	bar	True
f	-0.698294	0.484663	1.120755	bar	False
g	NaN	NaN	NaN	NaN	NaN
h	-2.119560	0.173416	0.700130	bar	False

1.0.51 对含有缺失值的列进行计数，求非缺失值的个数。

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

[2]:
```
# 生成 dataframe
df = pd.DataFrame(
    np.random.randn(5, 3),
    index=["a", "c", "e", "f", "h"],
    columns=["one", "two", "three"],
)
df["four"] = "bar"
df["five"] = df["one"] > 0
df = df.reindex(["a", "b", "c", "d", "e", "f", "g", "h"])
df
```

[2]:

	one	two	three	four	five
a	-0.126636	-1.491391	0.700825	bar	False
b	NaN	NaN	NaN	NaN	NaN
c	0.981905	-1.506363	0.082841	bar	True
d	NaN	NaN	NaN	NaN	NaN
e	-0.383442	1.491860	-0.551062	bar	False
f	-0.182666	0.236521	0.678531	bar	False
g	NaN	NaN	NaN	NaN	NaN
h	0.984269	-0.652942	-1.957255	bar	True

[3]:
```
# 求非缺失值的个数
df["one"].count()
```

[3]:
```
5
```

1.0.52 对含有缺失值的列进行插值。

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

[2]:
```
# 生成 dataframe
df = pd.DataFrame(
    np.random.randn(5, 3),
    index=["a", "c", "e", "f", "h"],
    columns=["one", "two", "three"],
)
df["four"] = "bar"
df["five"] = df["one"] > 0
df = df.reindex(["a", "b", "c", "d", "e", "f", "g", "h"])
df
```

[2]:

	one	two	three	four	five
a	0.957477	0.356867	1.238439	bar	True
b	NaN	NaN	NaN	NaN	NaN
c	-1.194969	-0.859723	-2.106189	bar	False
d	NaN	NaN	NaN	NaN	NaN

```
e  0.327321  0.633526 -0.024933  bar   True
f  1.027312  0.586582  0.242355  bar   True
g       NaN       NaN       NaN  NaN    NaN
h  0.095649  0.213300  0.137504  bar   True
```

[3]:
```
# 线性插值
df["one"].interpolate()
```

[3]:
```
a    0.957477
b   -0.118746
c   -1.194969
d   -0.433824
e    0.327321
f    1.027312
g    0.561480
h    0.095649
Name: one, dtype: float64
```

1.0.53 以 openml 中的数据集 1StudentPerformance 为例，将 gender 变量变为 category 类型。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
```

```
3                    none          47              57              44
4                    none          76              78              75
```

[4]:
```
# 将字符串变量 category 化
data["gender"].astype("category")
```

[4]:
```
0      female
1      female
2      female
3        male
4        male
       ...
995    female
996      male
997    female
998    female
999    female
Name: gender, Length: 1000, dtype: category
Categories (2, object): ['female', 'male']
```

1.0.54 以 openml 中的数据集 1StudentPerformance 为例，将 gender 变量变为 category 类型并修改类别名称为”男性”和”女性”。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
```

0	none	72	72	74
1	completed	69	90	88
2	none	90	95	93
3	none	47	57	44
4	none	76	78	75

[4]:
```
# 将 gender 修改为分类变量
newvar = data["gender"].astype("category")
newvar.head()
```

[4]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (2, object): ['female', 'male']
```

[5]:
```
# 查看分类变量的类别名称
print(newvar.cat.categories)
```

```
Index(['female', 'male'], dtype='object')
```

[6]:
```
# 修改分类变量的类别名称
new_categories = ["女性", "男性"]
newvar = newvar.cat.rename_categories(new_categories)
newvar.head()
```

[6]:
```
0    女性
1    女性
2    女性
3    男性
4    男性
Name: gender, dtype: category
Categories (2, object): ['女性', '男性']
```

1.0.55 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **gender** 变量变为 **category** 类型并添加一个类别”**Unknown**”。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
```

```
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 将 gender 修改为分类变量
newvar = data["gender"].astype("category")
newvar.head()
```

[4]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (2, object): ['female', 'male']
```

[5]:
```
# 添加一个类别
newvar = newvar.cat.add_categories(["Unknown"])
newvar.head()
```

[5]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (3, object): ['female', 'male', 'Unknown']
```

1.0.56 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **gender** 变量变为 **category** 类型并添加一个类别**"Unknown"**，然后删除该类别。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 将 gender 修改为分类变量
newvar = data["gender"].astype("category")
newvar.head()
```

[4]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (2, object): ['female', 'male']
```

[5]:
```
# 添加一个类别
newvar = newvar.cat.add_categories(["Unknown"])
newvar.head()
```

[5]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (3, object): ['female', 'male', 'Unknown']
```

[6]:
```
newvar.cat.remove_categories(["Unknown"])
```

[6]:
```
0      female
1      female
2      female
3        male
4        male
        …
995    female
996      male
997    female
998    female
999    female
Name: gender, Length: 1000, dtype: category
Categories (2, object): ['female', 'male']
```

[7]:
```
newvar.cat.remove_categories(["Unknown", "female"])
```

[7]:
```
0       NaN
1       NaN
2       NaN
3      male
4      male
       …
995     NaN
996    male
997     NaN
998     NaN
999     NaN
Name: gender, Length: 1000, dtype: category
Categories (1, object): ['male']
```

1.0.57 以 **openml** 中的数据集 **1StudentPerformance** 为例，将 **gender** 变量变为 **category** 类型并添加一个类别**"Unknown"**，然后删除没用的类别（类别无变量值对应）。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A          associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 将 gender 修改为分类变量
newvar = data["gender"].astype("category")
newvar.head()
```

[4]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (2, object): ['female', 'male']
```

[5]:
```
# 添加一个类别
newvar = newvar.cat.add_categories(["Unknown"])
newvar.head()
```

[5]:
```
0    female
1    female
2    female
3      male
4      male
Name: gender, dtype: category
Categories (3, object): ['female', 'male', 'Unknown']
```

[6]:
```
# 删除无用类别
newvar = newvar.cat.remove_unused_categories()
newvar
```

[6]:
```
0      female
1      female
2      female
3        male
4        male
        ...
995    female
996      male
997    female
998    female
999    female
Name: gender, Length: 1000, dtype: category
Categories (2, object): ['female', 'male']
```

1.0.58 以 openml 中的数据集 1StudentPerformance 为例，将变量 parental level of education 变为有序的 category 类型。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
```

```
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 将 gender 修改为分类变量
t = pd.CategoricalDtype([
    "some high school","high school","some college",
    "associate\\'s degree","bachelor\\'s degree","master\\'s degree"
    ],
    ordered=True # 从小打到排序
)
newvar = data["parental level of education"].astype(t)
newvar.head()
```

[4]:
```
0     bachelor\'s degree
1           some college
2       master\'s degree
3    associate\'s degree
4           some college
Name: parental level of education, dtype: category
Categories (6, object): ['some high school' < 'high school' < 'some college' <
'associate\'s degree' < 'bachelor\'s degree' < 'master\'s degree']
```

1.0.59 以 **openml** 中的数据集 **1StudentPerformance** 为例，将变量 **parental level of education** 变为有序的 **category** 类型并且修改类别之间的顺序。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C               some college      standard
2  female        group B            master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C               some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 将 gender 修改为分类变量
t = pd.CategoricalDtype([
    "some high school","high school","some college",
    "associate\\'s degree","bachelor\\'s degree","master\\'s degree"
    ],
    ordered=True # 从小打到排序
)
newvar = data["parental level of education"].astype(t)
newvar.head()
```

[4]:
```
0     bachelor\'s degree
1           some college
2       master\'s degree
3    associate\'s degree
4           some college
Name: parental level of education, dtype: category
Categories (6, object): ['some high school' < 'high school' < 'some college' <
'associate\'s degree' < 'bachelor\'s degree' < 'master\'s degree']
```

[6]:
```
newvar = data["parental level of education"].cat.reorder_categories([
    "high school","some high school","associate\\'s degree",
    "some college","bachelor\\'s degree","master\\'s degree"
    ],
    ordered=True # 从小打到排序
)
newvar.head()
```

[6]:
```
0     bachelor\'s degree
1           some college
```

```
2       master\'s degree
3    associate\'s degree
4           some college
Name: parental level of education, dtype: category
Categories (6, object): ['high school' < 'some high school' < 'associate\'s
degree' < 'some college' < 'bachelor\'s degree' < 'master\'s degree']
```

1.0.60 以 **openml** 中的数据集 **1StudentPerformance** 为例，将变量 **parental level of education** 变为有序的 **category** 类型并且对该列和 **math score** 作为因子排序。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:

	gender	race/ethnicity	parental level of education	lunch	\
0	female	group B	bachelor\'s degree	standard	
1	female	group C	some college	standard	
2	female	group B	master\'s degree	standard	
3	male	group A	associate\'s degree	free/reduced	
4	male	group C	some college	standard	

	test preparation course	math score	reading score	writing score
0	none	72	72	74
1	completed	69	90	88
2	none	90	95	93
3	none	47	57	44
4	none	76	78	75

[4]:
```
# 将 gender 修改为分类变量
t = pd.CategoricalDtype([
    "some high school","high school","some college",
    "associate\\'s degree","bachelor\\'s degree","master\\'s degree"
    ],
    ordered=True # 从小打到排序
)
```

```
data["parental level of education"] = data["parental level of education"].astype(t)
data["parental level of education"].head()
```

[4]:
```
0     bachelor\'s degree
1           some college
2       master\'s degree
3    associate\'s degree
4           some college
Name: parental level of education, dtype: category
Categories (6, object): ['some high school' < 'high school' < 'some college' <
'associate\'s degree' < 'bachelor\'s degree' < 'master\'s degree']
```

[5]:
```
# 排序
data.sort_values(by=["parental level of education", "math score"], ascending=False)
```

[5]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
618	male	group D	master\'s degree	standard	none	95	81	84
685	female	group E	master\'s degree	standard	completed	94	99	100
957	female	group D	master\'s degree	standard	none	92	100	100
846	male	group C	master\'s degree	standard	completed	91	85	85
2	female	group B	master\'s degree	standard	none	90	95	93
..	…	…	…	…	…	…	…	…
683	female	group C	some high school	free/reduced	completed	29	40	44
363	female	group D	some high school	free/reduced	none	27	34	32
338	female	group B	some high school	free/reduced	none	24	38	27
17	female	group B	some high school	free/reduced	none	18	32	28
59	female	group C	some high school	free/reduced	none	0	17	10

[1000 rows x 8 columns]

2 强化技能

2.0.1 以 **openml** 中的数据集 **1StudentPerformance** 为例，新建一列 **id** 表示不同学生的编号，创建一个矩阵，矩阵的第 i 行第 j 列是第 i 个学生的数学成绩 **math score** 减去第 j 个学生的数学成绩的结果。使用 **for** 循环对 **dataframe** 的元素一个一个赋值，赋值方式使用 **loc** 和 **at**，对比时间消耗。为了节省时间，限制 $1 <= i, j <= 200$。

[1]:
```
# 导入数据集获取工具
from sklearn.datasets import fetch_openml
# 导入数据分析库
import pandas as pd
```

[2]:
```
# 获取数据
data = fetch_openml(
    data_id=43255,
    as_frame=True,
    parser="pandas"
)["frame"]
```

[3]:
```
# 查看数据集的前五行
data.head()
```

[3]:
```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B          bachelor\'s degree      standard
1  female        group C                some college      standard
2  female        group B             master\'s degree      standard
3    male        group A         associate\'s degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
# 新建 id 列
data["id"] = ["ID{}".format(i) for i in range(1, data.shape[0]+1)]
```

[5]:
```
%%timeit
# 新建一个 dataframe
resdf1 = pd.DataFrame(columns=data["id"], index=data["id"])
for i in range(1, 201):
    for j in range(1, 201):
        resdf1.at["ID{}".format(i), "ID{}".format(j)] = data.at[i-1, "math score"]␣
↪- data.at[j-1, "math score"]
```

```
2.66 s ± 95.1 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

[6]:
```
%%timeit
# 新建一个 dataframe
resdf2 = pd.DataFrame(columns=data["id"], index=data["id"])
for i in range(1, 201):
    for j in range(1, 201):
        resdf2.loc["ID{}".format(i), "ID{}".format(j)] = data.loc[i-1, "math␣
↪score"] - data.loc[j-1, "math score"]
```

```
6.97 s ± 270 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```