第五部分 Pandas数据预处理（鸢尾花数据处理）

5-1 合并数据

```
# 生成数据
# 创建DataFrame
import pandas as pd
df1 = pd.DataFrame({"Name":["张三","李四","王五"],
                    "Age":[18,25,30],
                    "Sex":["男","女","女"]})
df1
```

	Name	Age	Sex
0	张三	18	男
1	李四	25	女
2	王五	30	女

```
df2 = pd.DataFrame({"Name":["赵六","钱七"],
                    "Age":[19,21],
                    "Sex":["男","女"]})
df2
```

	Name	Age	Sex
0	赵六	19	男
1	钱七	21	女

```
# 纵向合并
df = pd.concat([df1,df2], axis=0, ignore_index=True)
df
```

	Name	Age	Sex
0	张三	18	男
1	李四	25	女
2	王五	30	女
3	赵六	19	男
4	钱七	21	女

```
# 生成语文数学成绩
dict = {"chn":[90,85,62,58,75],
        "math":[96,66,73,60,90]}
df3 = pd.DataFrame(dict)
df3
```

	chn	math
0	90	96
1	85	66
2	62	73
3	58	60
4	75	90

```
# 横向合并
df_all = pd.concat([df,df3],axis=1)
df_all
```

	Name	Age	Sex	chn	math
0	张三	18	男	90	96
1	李四	25	女	85	66
2	王五	30	女	62	73
3	赵六	19	男	58	60
4	钱七	21	女	75	90

项目步骤：学生成绩和班级表的合并

```
# 基于键的合并
import pandas as pd
# 读入成绩表
df_score = pd.read_excel("data/sample.xlsx", sheet_name="Sheet3")
df_score
```

	姓名	班级编号	成绩
0	张三	1001	89
1	李四	1002	97
2	王五	1003	100
3	赵六	1001	60

```
# 读入班级表
df_class = pd.read_excel("data/sample.xlsx", sheet_name="Sheet4")
df_class
```

	班级号	班级名称
0	1001	计信1班
1	1002	计信2班
2	1003	计信3班

```
# 以班级编号为键合并
df_all = pd.merge(df_score,df_class,left_on="班级编号",right_on="班级号")
df_all
```

	姓名	班级编号	成绩	班级号	班级名称
0	张三	1001	89	1001	计信1班
1	赵六	1001	60	1001	计信1班
2	李四	1002	97	1002	计信2班
3	王五	1003	100	1003	计信3班

```
# 如果使用join函数，则需要两个表主键名字相同
df_class1 = pd.read_excel("data/sample.xlsx", sheet_name="Sheet6")
df_class1
```

	班级编号	班级名称
0	1001	计信1班
1	1002	计信2班
2	1003	计信3班

```
df_score.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   姓名      4 non-null      object
 1   班级编号    4 non-null      int64
 2   成绩      4 non-null      int64
dtypes: int64(2), object(1)
memory usage: 224.0+ bytes
```

```
df_all1 = df_score.join(df_class1, how="inner", rsuffix="class")
df_all1
```

	姓名	班级编号	成绩	班级编号class	班级名称
0	张三	1001	89	1001	计信1班
1	李四	1002	97	1002	计信2班
2	王五	1003	100	1003	计信3班

5-2 数据清洗

```
# 去除重复样本
import pandas as pd
df1 = pd.read_excel("data/sample.xlsx", sheet_name="Sheet1")
df2 = pd.read_excel("data/sample.xlsx", sheet_name="Sheet5")
df = pd.concat([df1,df2], ignore_index=True)
df
```

	ind	A	B	C	D
0	1	a0	b0	c0	d0
1	2	a1	b1	c1	d1
2	3	a2	b2	c2	d2
3	4	a4	b4	c4	d4
4	5	a5	b5	c5	d5
5	4	a4	b4	c4	d4
6	5	a5	b5	c5	d5
7	6	a6	b6	c6	d6
8	7	a7	b7	c7	d7

```
df1 = df.drop_duplicates()
df1
```

	ind	A	B	C	D
0	1	a0	b0	c0	d0
1	2	a1	b1	c1	d1
2	3	a2	b2	c2	d2
3	4	a4	b4	c4	d4
4	5	a5	b5	c5	d5
7	6	a6	b6	c6	d6
8	7	a7	b7	c7	d7

项目步骤：鸢尾花数据缺失值处理

```
# 读入鸢尾花数据
iris = pd.read_csv("data/iris-data.csv")
iris.head()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
# 发现缺失值
iris.isnull().sum()
```

```
sepal_length_cm    0
sepal_width_cm     0
petal_length_cm    0
petal_width_cm     5
class              0
dtype: int64
```

```
# 或者
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   sepal_length_cm  150 non-null    float64
 1   sepal_width_cm   150 non-null    float64
 2   petal_length_cm  150 non-null    float64
 3   petal_width_cm   145 non-null    float64
 4   class            150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
# 缺失值所在行
iris.loc[iris["petal_width_cm"].isnull(),]
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
7	5.0	3.4	1.5	NaN	Iris-setosa
8	4.4	2.9	1.4	NaN	Iris-setosa
9	4.9	3.1	1.5	NaN	Iris-setosa
10	5.4	3.7	1.5	NaN	Iris-setosa
11	4.8	3.4	1.6	NaN	Iris-setosa

```
# 缺失值处理
# 1、删除缺失值
iris.dropna(axis=0, how="any").info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 145 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   sepal_length_cm  145 non-null    float64
 1   sepal_width_cm   145 non-null    float64
 2   petal_length_cm  145 non-null    float64
 3   petal_width_cm   145 non-null    float64
 4   class            145 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.8+ KB
```

```
# 2、填充缺失值（前向、后向）
iris.fillna(method="bfill").info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sepal_length_cm  150 non-null    float64
 1   sepal_width_cm   150 non-null    float64
 2   petal_length_cm  150 non-null    float64
 3   petal_width_cm   150 non-null    float64
 4   class            150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
# 3、填充具体值
#iris.fillna(-99)
m = iris.loc[iris["class"]=="Iris-setosa", "petal_width_cm"].mean()
iris = iris.fillna(m)
iris.info()
```

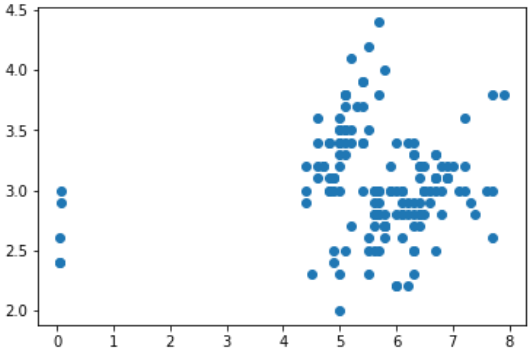
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sepal_length_cm  150 non-null    float64
 1   sepal_width_cm   150 non-null    float64
 2   petal_length_cm  150 non-null    float64
 3   petal_width_cm   150 non-null    float64
 4   class            150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

项目步骤：鸢尾花数据异常值处理

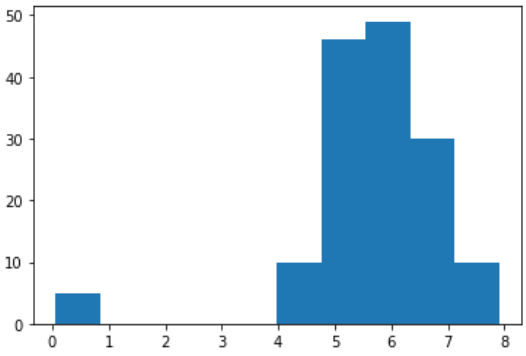
```
# 发现异常
iris.describe()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm
count	150.000000	150.000000	150.000000	150.000000
mean	5.644627	3.054667	3.758667	1.203667
std	1.312781	0.433123	1.764420	0.763252
min	0.055000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.700000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

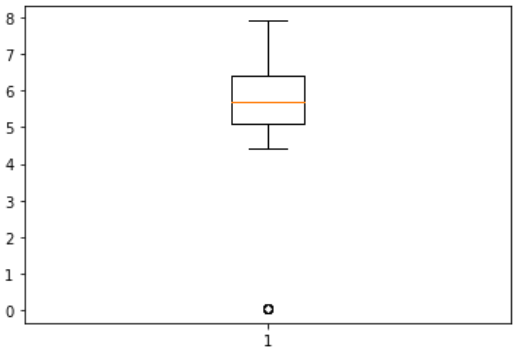
```
# 绘图方式发现异常值
import matplotlib.pyplot as plt
plt.scatter(iris.sepal_length_cm,iris.sepal_width_cm)
plt.show()
```




```
# 直方图发现异常
plt.hist(iris.sepal_length_cm)
plt.show()
```




```
# 箱型图发现异常
plt.boxplot(iris.sepal_length_cm)
plt.show()
```




```
# 定位异常位置
iris.loc[iris.sepal_length_cm < 1.0,:]
```

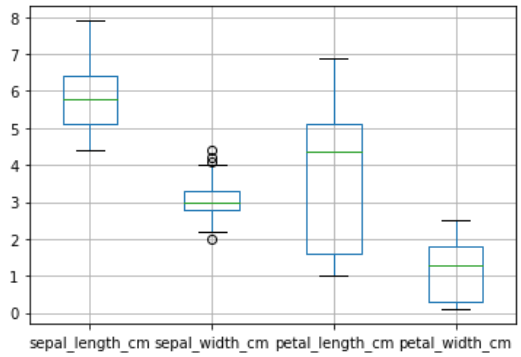
	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
77	0.067	3.0	5.0	1.7	Iris-versicolor
78	0.060	2.9	4.5	1.5	Iris-versicolor
79	0.057	2.6	3.5	1.0	Iris-versicolor
80	0.055	2.4	3.8	1.1	Iris-versicolor
81	0.055	2.4	3.7	1.0	Iris-versicolor

```
# 处理异常，单位厘米和米的换算
iris.loc[iris.sepal_length_cm < 1.0,"sepal_length_cm"] *= 100
iris.iloc[77:82,:]
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
77	6.7	3.0	5.0	1.7	Iris-versicolor
78	6.0	2.9	4.5	1.5	Iris-versicolor
79	5.7	2.6	3.5	1.0	Iris-versicolor
80	5.5	2.4	3.8	1.1	Iris-versicolor
81	5.5	2.4	3.7	1.0	Iris-versicolor

```
#iris = pd.read_csv("data/iris-data.csv")
iris.boxplot()
```

```
<AxesSubplot:>
```




```
# 类别异常
iris["class"].value_counts()
```

```
Iris-virginica     50
Iris-setosa        49
Iris-versicolor    45
versicolor          5
Iris-setossa        1
Name: class, dtype: int64
```

```
# 处理异常
iris.loc[iris["class"]=="Iris-setossa","class"] = "Iris-setosa"
iris.loc[iris["class"]=="versicolor","class"] = "Iris-versicolor"
iris["class"].value_counts()
```

```
Iris-setosa        50
Iris-virginica     50
Iris-versicolor    50
Name: class, dtype: int64
```

```
iris.to_csv("data/iris.csv",index=False)
```

5-3 数据标准化

项目步骤：鸢尾花数据标准化

```
# 读入清洗后的鸢尾花数据
import pandas as pd
iris = pd.read_csv("data/iris.csv")
iris.head()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sepal_length_cm  150 non-null    float64
 1   sepal_width_cm   150 non-null    float64
 2   petal_length_cm  150 non-null    float64
 3   petal_width_cm   150 non-null    float64
 4   class            150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
# 转换数据类型
iris.iloc[:,0:4] = iris.iloc[:,0:4].values.astype("float")
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sepal_length_cm  150 non-null    float64
 1   sepal_width_cm   150 non-null    float64
 2   petal_length_cm  150 non-null    float64
 3   petal_width_cm   150 non-null    float64
 4   class            150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
# 离差标准化数据
# 离差标准化函数
def max_min_scale(data):
    return (data-data.min())/(data.max()-data.min())
```

```
# 对花瓣长度进行离差标准化
iris["sepal_length_cm"] = max_min_scale(iris["sepal_length_cm"])
iris.head()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
0	0.200000	3.5	1.4	0.2	Iris-setosa
1	0.142857	3.0	1.4	0.2	Iris-setosa
2	0.085714	3.2	1.3	0.2	Iris-setosa
3	0.057143	3.1	1.5	0.2	Iris-setosa
4	0.171429	3.6	1.4	0.2	Iris-setosa

```
# 标准差标准化数据
def std_scale(data):
    return (data-data.mean())/data.std()
```

```
# 对花瓣宽度进行离差标准化
iris["sepal_width_cm"] = std_scale(iris["sepal_width_cm"])
iris.head()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
0	0.200000	1.028191	1.4	0.2	Iris-setosa
1	0.142857	-0.126215	1.4	0.2	Iris-setosa
2	0.085714	0.335547	1.3	0.2	Iris-setosa
3	0.057143	0.104666	1.5	0.2	Iris-setosa
4	0.171429	1.259073	1.4	0.2	Iris-setosa

```
# 小数定标标准化数据
import numpy as np
##自定义小数定标差标准化函数
def decimal_scale(data):
    return data/10**np.ceil(np.log10(data.abs().max()))
```

```
# 对花萼长度进行离差标准化
iris["petal_length_cm"] = decimal_scale(iris["petal_length_cm"])
iris.head()
```

	sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	class
0	0.200000	1.028191	0.14	0.2	Iris-setosa
1	0.142857	-0.126215	0.14	0.2	Iris-setosa
2	0.085714	0.335547	0.13	0.2	Iris-setosa
3	0.057143	0.104666	0.15	0.2	Iris-setosa
4	0.171429	1.259073	0.14	0.2	Iris-setosa

5-4 数据转换

项目步骤：学生成绩数据的转换

```
# 读入数据
import pandas as pd
scores = pd.read_csv("data/scores.csv", encoding="gbk")
scores.head()
```

	num	class	chn	math	eng	phy	chem	politics	bio	history	geo	pe	total	gender
0	158	3	99.0	120	114.0	70.0	49.50	50.0	49.0	48.5	49.5	60	709.50	女
1	442	7	107.0	120	118.5	68.6	43.00	49.0	48.5	48.5	49.0	56	708.10	男
2	249	4	98.0	120	116.0	70.0	47.50	47.0	49.0	47.5	49.0	60	704.00	男
3	573	9	102.0	113	111.5	70.0	47.00	49.0	49.0	49.0	49.5	60	700.00	女
4	310	5	103.0	120	111.5	70.0	44.75	46.5	48.0	48.0	48.0	60	699.75	女

```
# 性别的哑变量处理
genders = pd.get_dummies(scores["gender"], prefix="sex")
scores = pd.concat([scores,genders],axis=1)
scores.drop("gender",axis=1,inplace=True)
scores.head()
```

	num	class	chn	math	eng	phy	chem	politics	bio	history	geo	pe	total	sex_女	se
0	158	3	99.0	120	114.0	70.0	49.50	50.0	49.0	48.5	49.5	60	709.50	1	0
1	442	7	107.0	120	118.5	68.6	43.00	49.0	48.5	48.5	49.0	56	708.10	0	1
2	249	4	98.0	120	116.0	70.0	47.50	47.0	49.0	47.5	49.0	60	704.00	0	1
3	573	9	102.0	113	111.5	70.0	47.00	49.0	49.0	49.0	49.5	60	700.00	1	0
4	310	5	103.0	120	111.5	70.0	44.75	46.5	48.0	48.0	48.0	60	699.75	1	0

```
genders.head()
```

	sex_女	sex_男
0	1	0
1	0	1
2	0	1
3	1	0
4	1	0

```
# 对英语成绩离散化
eng_grade = pd.cut(scores["eng"],bins=[0,72,84,96,108,120],labels=["不及格","及格","中","良","优"])
scores["eng"] = eng_grade
scores.head()
```

	num	class	chn	math	eng	phy	chem	politics	bio	history	geo	pe	total	sex_女	sex_男
0	158	3	99.0	120	优	70.0	49.50	50.0	49.0	48.5	49.5	60	709.50	1	0
1	442	7	107.0	120	优	68.6	43.00	49.0	48.5	48.5	49.0	56	708.10	0	1
2	249	4	98.0	120	优	70.0	47.50	47.0	49.0	47.5	49.0	60	704.00	0	1
3	573	9	102.0	113	优	70.0	47.00	49.0	49.0	49.0	49.5	60	700.00	1	0
4	310	5	103.0	120	优	70.0	44.75	46.5	48.0	48.0	48.0	60	699.75	1	0

```
scores.tail()
```

	num	class	chn	math	eng	phy	chem	politics	bio	history	geo	pe	total	sex_女	se
594	509	8	64.0	20	不及格	14.7	12.00	24.0	17.0	11.0	16.0	48	249.70	1	0
595	244	4	26.5	35	不及格	9.8	8.25	31.5	21.5	8.5	26.0	48	230.55	0	1
596	335	6	29.0	18	不及格	10.5	17.25	17.0	20.5	15.5	27.5	52	225.25	1	0
597	131	3	48.5	22	不及格	10.5	12.25	15.0	17.5	13.0	10.5	48	215.75	1	0
598	25	1	36.0	9	不及格	21.7	11.50	23.5	15.0	8.0	11.5	48	214.20	1	0

```

```