

1. 创建宿舍成员基本信息表(DataFrame)。

源程序:

```
import pandas as pd
```

```
# 创建数据
```

```
data = {  
    '编号': [1, 2, 3, 4, 5, 6],  
    '姓名': ['郭康欣', '边水柔', '崔以墨', '耿雨菲', '郭硕', '占位'],  
    '学号': ['2305014208', '2305014202', '2305014204', '2305014207', '2305014209', '2300000000'],  
    '性别': ['女', '女', '女', '女', '女', '女'],  
    '年龄': [19, 19, 19, 19, 19, 19],  
    '出生年月日': ['2004-01-15', '2004-03-22', '2004-05-10', '2004-07-18', '2004-09-25', '2004-01-01'],  
    '身高': [165, 160, 158, 162, 167, 160],  
    '体重': [50, 48, 47, 52, 55, 60],  
    '爱好': ['阅读', '跑步', '音乐', '绘画', '篮球', '占位'],  
    '宿舍号': [101, 101, 101, 101, 101, 101]  
}
```

```
# 创建 DataFrame
```

```
df = pd.DataFrame(data)
```

```
display(df)
```

过程性结果:

```
[3]: import pandas as pd

# 创建数据
data = {
    '编号': [1, 2, 3, 4, 5, 6],
    '姓名': ['郭康欣', '边水柔', '崔以墨', '耿雨菲', '郭硕', '占位'],
    '学号': ['2305014208', '2305014202', '2305014204', '2305014207', '2305014209', '2300000000'],
    '性别': ['女', '女', '女', '女', '女', '女'],
    '年龄': [19, 19, 19, 19, 19, 19],
    '出生年月日': ['2004-01-15', '2004-03-22', '2004-05-10', '2004-07-18', '2004-09-25', '2004-01-01'],
    '身高': [165, 160, 158, 162, 167, 160],
    '体重': [50, 48, 47, 52, 55, 60],
    '爱好': ['阅读', '跑步', '音乐', '绘画', '篮球', '占位'],
    '宿舍号': [101, 101, 101, 101, 101, 101]
}

# 创建DataFrame
df = pd.DataFrame(data)
display(df)
```

	编号	姓名	学号	性别	年龄	出生年月日	身高	体重	爱好	宿舍号
0	1	郭康欣	2305014208	女	19	2004-01-15	165	50	阅读	101
1	2	边水柔	2305014202	女	19	2004-03-22	160	48	跑步	101
2	3	崔以墨	2305014204	女	19	2004-05-10	158	47	音乐	101
3	4	耿雨菲	2305014207	女	19	2004-07-18	162	52	绘画	101
4	5	郭硕	2305014209	女	19	2004-09-25	167	55	篮球	101
5	6	占位	2300000000	女	19	2004-01-01	160	60	占位	101

2. 求该宿舍成员的身高、体重的均值、最大值、最小值、标准差等统计信息，并按出生年月日对该宿舍成员进行排序。

源程序：

```
# 计算统计信息
```

```
height_stats = df['身高'].astype(float).describe()
```

```
weight_stats = df['体重'].astype(float).describe()
```

```
print("身高统计信息：")
```

```
display(height_stats)
```

```
print("体重统计信息：")
```

```
display(weight_stats)
```

```
# 按出生年月日排序

sorted_df = df.sort_values(by='出生年月日')

display(sorted_df)
```

过程性结果：

```
[5]: # 计算统计信息
height_stats = df['身高'].astype(float).describe()
weight_stats = df['体重'].astype(float).describe()

print("身高统计信息：")
display(height_stats)

print("体重统计信息：")
display(weight_stats)

# 按出生年月日排序
sorted_df = df.sort_values(by='出生年月日')
display(sorted_df)
```

身高统计信息：

count	6.000000
mean	162.000000
std	3.405877
min	158.000000
25%	160.000000
50%	161.000000
75%	164.250000
max	167.000000

Name: 身高, dtype: float64

体重统计信息：

count	6.000000
mean	52.000000
std	4.857983
min	47.000000
25%	48.500000
50%	51.000000
75%	54.250000
max	60.000000

Name: 体重, dtype: float64

	编号	姓名	学号	性别	年龄	出生年月日	身高	体重	爱好	宿舍号
5	6	占位	2300000000	女	19	2004-01-01	160	60	占位	101
0	1	郭康欣	2305014208	女	19	2004-01-15	165	50	阅读	101
1	2	边水柔	2305014202	女	19	2004-03-22	160	48	跑步	101
2	3	崔以墨	2305014204	女	19	2004-05-10	158	47	音乐	101
3	4	耿雨菲	2305014207	女	19	2004-07-18	162	52	绘画	101
4	5	郭硕	2305014209	女	19	2004-09-25	167	55	篮球	101

3. 创建 2023-2024 学年第一学期宿舍成员科目成绩信息表

源程序：

```
# 创建成绩数据

grades = {

    '编号': [1, 2, 3, 4, 5, 6],

    '姓名': ['郭康欣', '边水柔', '崔以墨', '耿雨菲', '郭硕', '占位'],
```

```

'学号': ['2305014208', '2305014202', '2305014204', '2305014207', '2305014209', '2300000000'],
'数学分析(一)': [85, 90, 78, 88, 92, 90],
'解析几何': [88, 85, 80, 82, 89, 90],
'大学计算机': [90, 92, 85, 87, 91, 90],
'大学英语(1)': [87, 89, 83, 86, 88, 90],
'思想道德修养与法律基础': [86, 88, 81, 84, 87, 90]
}

```

创建 DataFrame

```
grades_df = pd.DataFrame(grades)
```

```
display(grades_df)
```

过程性结果：

```

[11]: # 创建成绩数据
grades = {
    '编号': [1, 2, 3, 4, 5, 6],
    '姓名': ['郭康欣', '边水柔', '崔以墨', '耿雨菲', '郭硕', '占位'],
    '学号': ['2305014208', '2305014202', '2305014204', '2305014207', '2305014209', '2300000000'],
    '数学分析(一)': [85, 90, 78, 88, 92, 90],
    '解析几何': [88, 85, 80, 82, 89, 90],
    '大学计算机': [90, 92, 85, 87, 91, 90],
    '大学英语(1)': [87, 89, 83, 86, 88, 90],
    '思想道德修养与法律基础': [86, 88, 81, 84, 87, 90]
}

# 创建DataFrame
grades_df = pd.DataFrame(grades)
display(grades_df)

```

	编号	姓名	学号	数学分析(一)	解析几何	大学计算机	大学英语(1)	思想道德修养与法律基础
0	1	郭康欣	2305014208	85	88	90	87	86
1	2	边水柔	2305014202	90	85	92	89	88
2	3	崔以墨	2305014204	78	80	85	83	81
3	4	耿雨菲	2305014207	88	82	87	86	84
4	5	郭硕	2305014209	92	89	91	88	87
5	6	占位	2300000000	90	90	90	90	90

4. 使用 isnull()对宿舍成员成绩信息表进行缺失值检测

源程序：

检测缺失值

```
missing_values = grades_df.isnull().sum()
```

```
print(missing_values)
```

过程性结果：

```
[12]: # 检测缺失值
missing_values = grades_df.isnull().sum()
print(missing_values)
```

```
编号          0
姓名          0
学号          0
数学分析(一)    0
解析几何        0
大学计算机      0
大学英语(1)    0
思想道德修养与法律基础  0
dtype: int64
```

5. 宿舍成员科目成绩求和，并追加到表最后一列，并按成绩进行排序

源程序：

```
# 计算总成绩
```

```
grades_df['总成绩'] = grades_df.iloc[:, 3:].sum(axis=1)
```

```
# 按总成绩排序
```

```
sorted_grades_df = grades_df.sort_values(by='总成绩', ascending=False)
```

```
display(sorted_grades_df)
```

过程性结果：

```
[15]: # 计算总成绩
grades_df['总成绩'] = grades_df.iloc[:, 3:].sum(axis=1)

# 按总成绩排序
sorted_grades_df = grades_df.sort_values(by='总成绩', ascending=False)
display(sorted_grades_df)
```

	编号	姓名	学号	数学分析(一)	解析几何	大学计算机	大学英语(1)	思想道德修养与法律基础	总成绩
5	6	占位	2300000000	90	90	90	90	90	900
4	5	郭硕	2305014209	92	89	91	88	87	894
1	2	边水柔	2305014202	90	85	92	89	88	888
0	1	郭康欣	2305014208	85	88	90	87	86	872
3	4	耿雨菲	2305014207	88	82	87	86	84	854
2	3	崔以墨	2305014204	78	80	85	83	81	814

6. 用 describe()方法对各科目成绩进行描述性统计

源程序：

描述性统计

stats = grades_df.describe()

display(stats)

过程性结果：

6. 用describe()方法对各科目成绩进行描述性统计

```
[17]: # 描述性统计
stats = grades_df.describe()
display(stats)
```

	编号	数学分析(一)	解析几何	大学计算机	大学英语(1)	思想道德修养与法律基础	总成绩
count	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000
mean	3.500000	87.166667	85.666667	89.166667	87.166667	86.000000	870.333333
std	1.870829	5.076088	4.033196	2.639444	2.483277	3.162278	32.234557
min	1.000000	78.000000	80.000000	85.000000	83.000000	81.000000	814.000000
25%	2.250000	85.750000	82.750000	87.750000	86.250000	84.500000	858.500000
50%	3.500000	89.000000	86.500000	90.000000	87.500000	86.500000	880.000000
75%	4.750000	90.000000	88.750000	90.750000	88.750000	87.750000	892.500000
max	6.000000	92.000000	90.000000	92.000000	90.000000	90.000000	900.000000

7. 使用 Matplotlib 库分别绘制数学分析（一）成绩的柱状图

源程序：

```
import matplotlib.pyplot as plt
```

```
plt.rcParams['font.sans-serif']=['SimHei'] # 用来正常显示中文标签（中文乱码问题）
```

```
# 绘制柱状图
```

```
plt.bar(grades_df['姓名'], grades_df['数学分析(一)'])
```

```
plt.xlabel('姓名')
```

```
plt.ylabel('数学分析(一)成绩')
```

```
plt.title('数学分析(一)成绩分布')
```

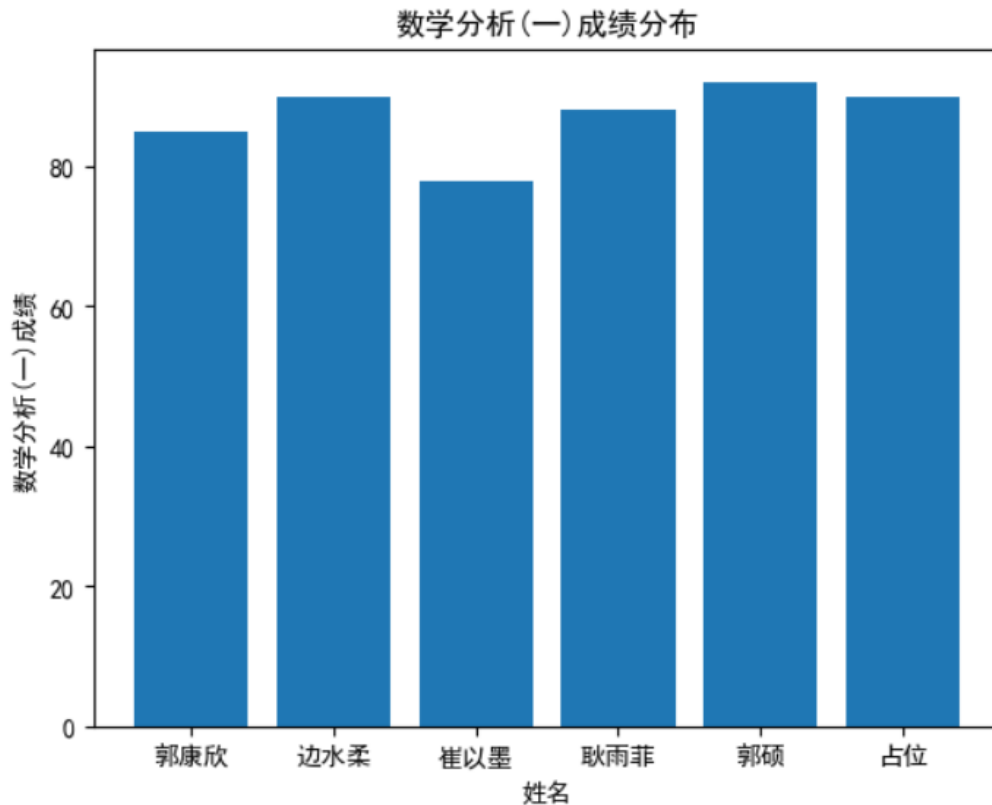
```
plt.show()
```

过程性结果：

```
[19]: import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif']=['SimHei'] # 用来正常显示中文标签 (中文乱码问题)

# 绘制柱状图
plt.bar(grades_df['姓名'], grades_df['数学分析(一)'])
plt.xlabel('姓名')
plt.ylabel('数学分析(一)成绩')
plt.title('数学分析(一)成绩分布')
plt.show()
```



8. 使用 Matplotlib 库分别绘制数学分析（一）成绩的饼图

源程序：

绘制饼图

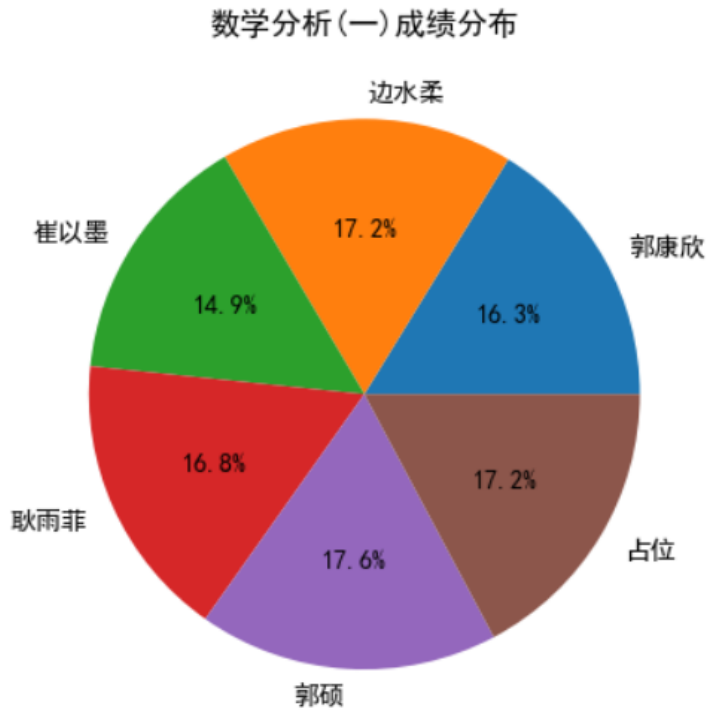
```
plt.pie(grades_df['数学分析(一)'], labels=grades_df['姓名'], autopct='%1.1f%%')
```

```
plt.title('数学分析(一)成绩分布')
```

```
plt.show()
```

过程性结果：


```
[21]: # 绘制饼图
plt.pie(grades_df['数学分析(一)'], labels=grades_df['姓名'], autopct='%1.1f%%')
plt.title('数学分析(一)成绩分布')
plt.show()
```



9. 用箱线图分别对数学分析（一）成绩进行异常值检测

源程序：

```
# 绘制箱线图

plt.boxplot(grades_df['数学分析(一)'])

plt.xticks([1], ['数学分析(一)'])

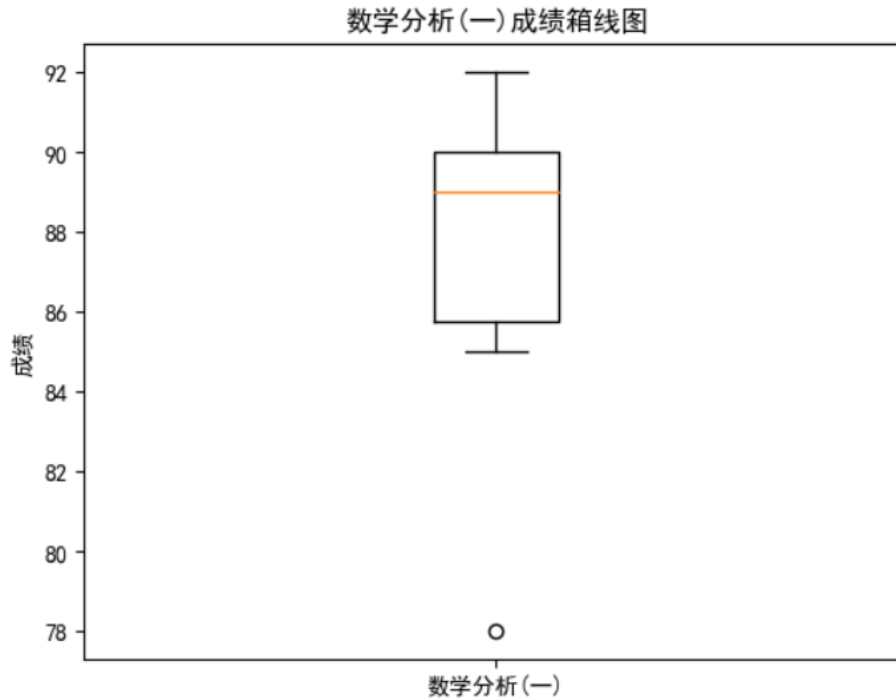
plt.ylabel('成绩')

plt.title('数学分析(一)成绩箱线图')

plt.show()
```

过程性结果：

```
[23]: # 绘制箱线图
plt.boxplot(grades_df['数学分析(一)'])
plt.xticks([1], ['数学分析(一)'])
plt.ylabel('成绩')
plt.title('数学分析(一)成绩箱线图')
plt.show()
```



10. 按姓名和学号将宿舍成员基本信息表和成绩信息表进行合并

源程序:

合并数据

```
merged_df = pd.merge(df, grades_df, on=['姓名', '学号'])
```

```
display(merged_df)
```

过程性结果:

10. 按姓名和学号将宿舍成员基本信息表和成绩信息表进行合并

```
[24]: # 合并数据
merged_df = pd.merge(df, grades_df, on=['姓名', '学号'])
display(merged_df)
```

	编号_x	姓名	学号	性别	年龄	出生年月日	身高	体重	爱好	宿舍号	编号_y	数学分析(一)	解析几何	大学计算机	大学英语(1)	思想道德修养与法律基础	总成绩
0	1	郭康欣	2305014208	女	19	2004-01-15	165	50	阅读	101	1	85	88	90	87	86	872
1	2	边水柔	2305014202	女	19	2004-03-22	160	48	跑步	101	2	90	85	92	89	88	888
2	3	崔以墨	2305014204	女	19	2004-05-10	158	47	音乐	101	3	78	80	85	83	81	814
3	4	耿雨菲	2305014207	女	19	2004-07-18	162	52	绘画	101	4	88	82	87	86	84	854
4	5	郭硕	2305014209	女	19	2004-09-25	167	55	篮球	101	5	92	89	91	88	87	894
5	6	占位	2300000000	女	19	2004-01-01	160	60	占位	101	6	90	90	90	90	90	900