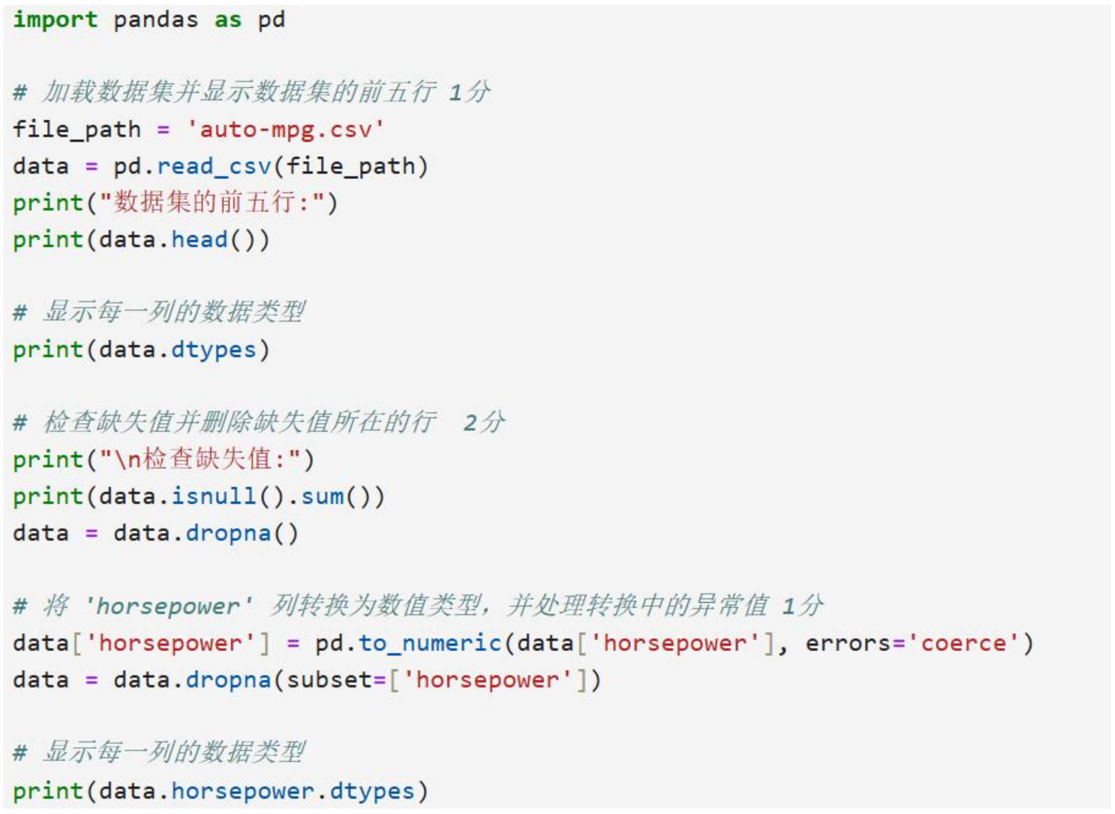
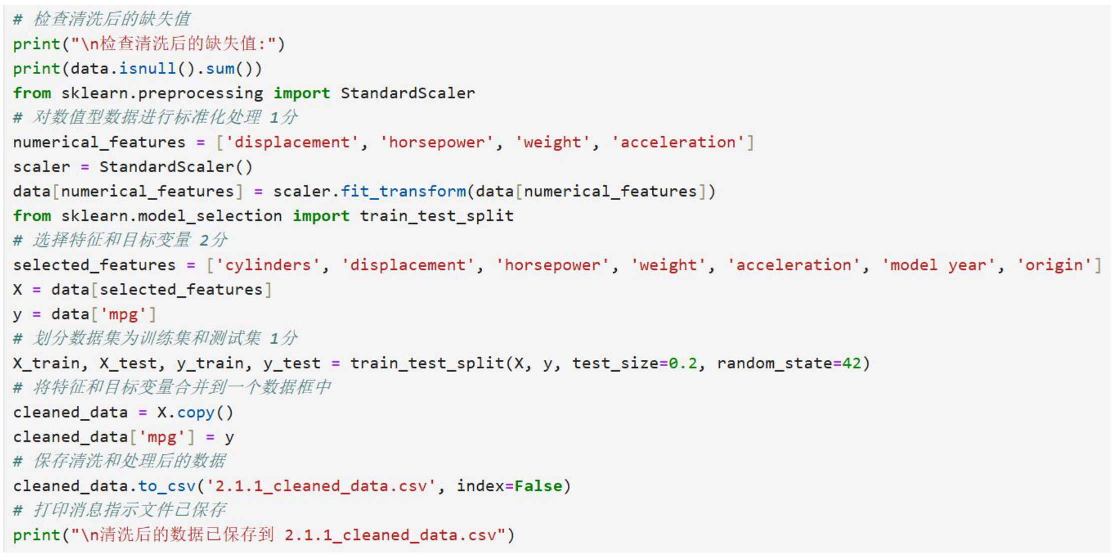
2.1.1 ipynb

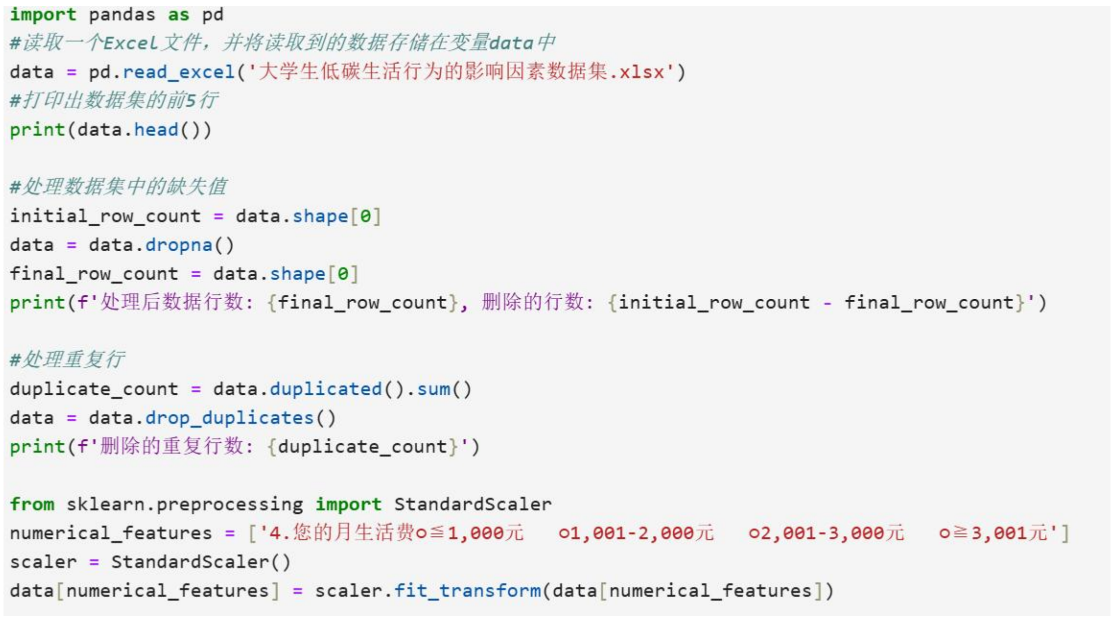


```python
import pandas as pd

# 加载数据集并显示数据集的前五行 1分
file_path = 'auto-mpg.csv'
data = pd.read_csv(file_path)
print("数据集的前五行:")
print(data.head())

# 显示每一列的数据类型
print(data.dtypes)

# 检查缺失值并删除缺失值所在的行  2分
print("\n检查缺失值:")
print(data.isnull().sum())
data = data.dropna()

# 将 'horsepower' 列转换为数值类型，并处理转换中的异常值 1分
data['horsepower'] = pd.to_numeric(data['horsepower'], errors='coerce')
data = data.dropna(subset=['horsepower'])

# 显示每一列的数据类型
print(data.horsepower.dtypes)
```

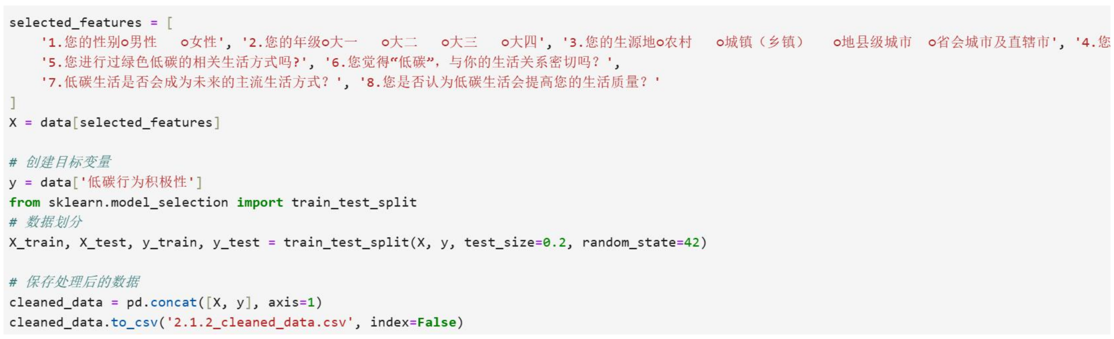


```python
# 检查清洗后的缺失值
print("\n检查清洗后的缺失值:")
print(data.isnull().sum())
from sklearn.preprocessing import StandardScaler
# 对数值型数据进行标准化处理 1分
numerical_features = ['displacement', 'horsepower', 'weight', 'acceleration']
scaler = StandardScaler()
data[numerical_features] = scaler.fit_transform(data[numerical_features])
from sklearn.model_selection import train_test_split
# 选择特征和目标变量 2分
selected_features = ['cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model year', 'origin']
X = data[selected_features]
y = data['mpg']
# 划分数据集为训练集和测试集 1分
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# 将特征和目标变量合并到一个数据框中
cleaned_data = X.copy()
cleaned_data['mpg'] = y
# 保存清洗和处理后的数据
cleaned_data.to_csv('2.1.1_cleaned_data.csv', index=False)
# 打印消息指示文件已保存
print("\n清洗后的数据已保存到 2.1.1_cleaned_data.csv")
```

2.1.2 ipynb

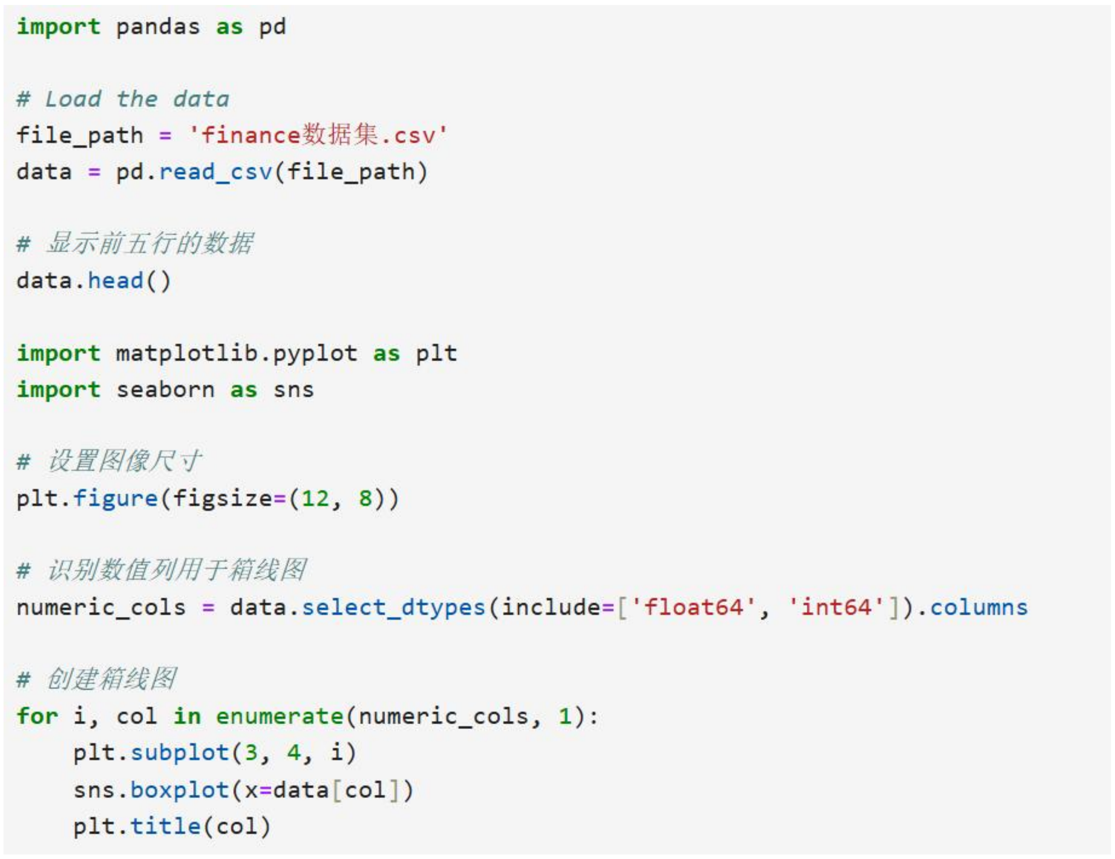


```
import pandas as pd
#读取一个Excel文件，并将读取到的数据存储在变量data中
data = pd.read_excel('大学生低碳生活行为的影响因素数据集.xlsx')
#打印出数据集的前5行
print(data.head())

#处理数据集中的缺失值
initial_row_count = data.shape[0]
data = data.dropna()
final_row_count = data.shape[0]
print(f'处理后数据行数: {final_row_count}, 删除的行数: {initial_row_count - final_row_count}')

#处理重复行
duplicate_count = data.duplicated().sum()
data = data.drop_duplicates()
print(f'删除的重复行数: {duplicate_count}')

from sklearn.preprocessing import StandardScaler
numerical_features = ['4.您的月生活费○≦1,000元    ○1,001-2,000元    ○2,001-3,000元    ○≧3,001元']
scaler = StandardScaler()
data[numerical_features] = scaler.fit_transform(data[numerical_features])
```

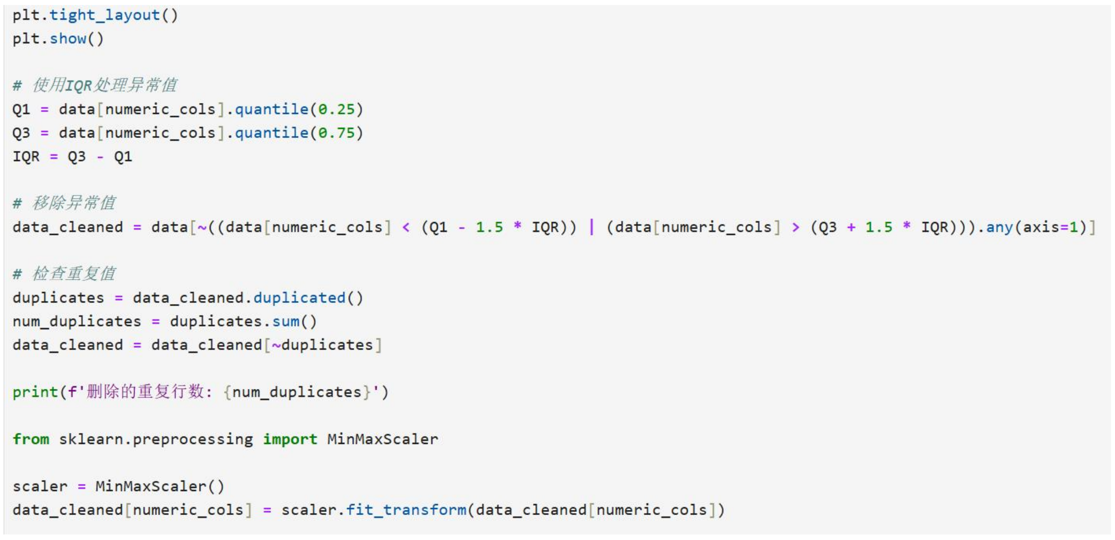


```
selected_features = [
    '1.您的性别○男性    ○女性', '2.您的年级○大一    ○大二    ○大三    ○大四', '3.您的生源地○农村    ○城镇（乡镇）    ○地县级城市    ○省会城市及直辖市', '4.您
    '5.您进行过绿色低碳的相关生活方式吗?', '6.您觉得“低碳”，与你的生活关系密切吗？',
    '7.低碳生活是否会成为未来的主流生活方式？', '8.您是否认为低碳生活会提高您的生活质量？'
]
X = data[selected_features]

# 创建目标变量
y = data['低碳行为积极性']
from sklearn.model_selection import train_test_split
# 数据划分
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 保存处理后的数据
cleaned_data = pd.concat([X, y], axis=1)
cleaned_data.to_csv('2.1.2_cleaned_data.csv', index=False)
```

2.1.3 ipynb



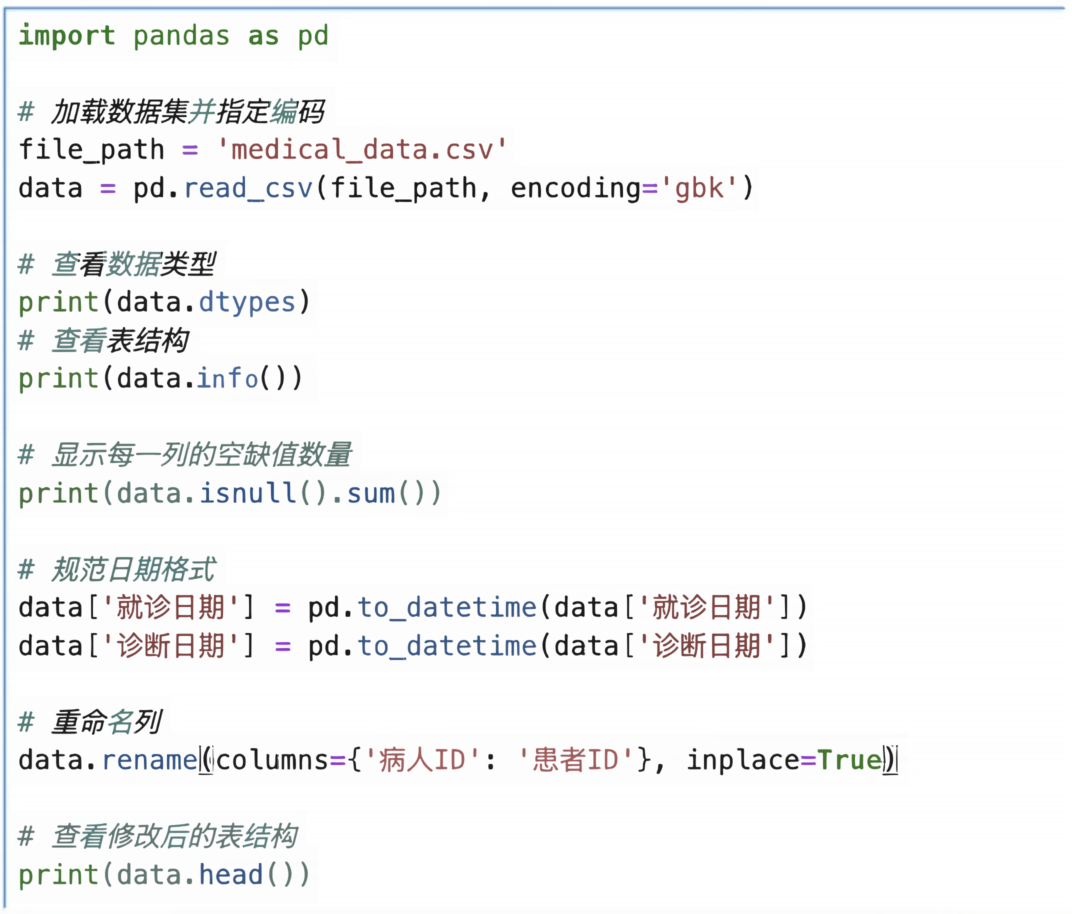
```python
import pandas as pd

# Load the data
file_path = 'finance数据集.csv'
data = pd.read_csv(file_path)

# 显示前五行的数据
data.head()

import matplotlib.pyplot as plt
import seaborn as sns

# 设置图像尺寸
plt.figure(figsize=(12, 8))

# 识别数值列用于箱线图
numeric_cols = data.select_dtypes(include=['float64', 'int64']).columns

# 创建箱线图
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(3, 4, i)
    sns.boxplot(x=data[col])
    plt.title(col)
```



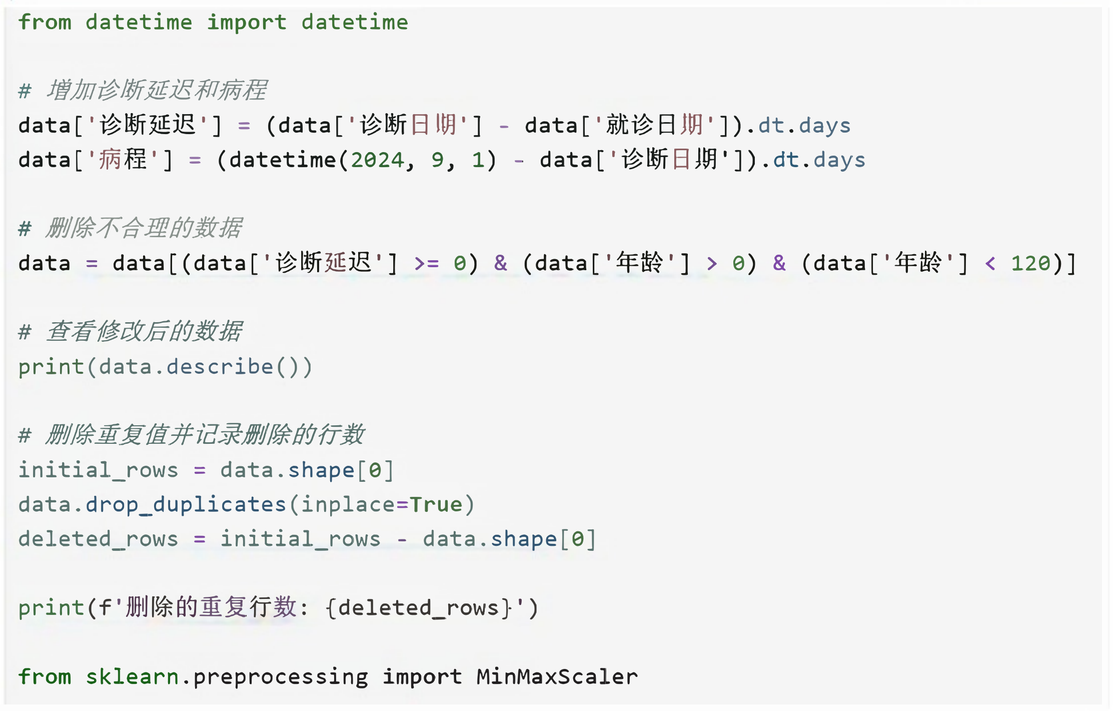
```python
plt.tight_layout()
plt.show()

# 使用IQR处理异常值
Q1 = data[numeric_cols].quantile(0.25)
Q3 = data[numeric_cols].quantile(0.75)
IQR = Q3 - Q1

# 移除异常值
data_cleaned = data[~((data[numeric_cols] < (Q1 - 1.5 * IQR)) | (data[numeric_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]

# 检查重复值
duplicates = data_cleaned.duplicated()
num_duplicates = duplicates.sum()
data_cleaned = data_cleaned[~duplicates]

print(f'删除的重复行数: {num_duplicates}')

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
data_cleaned[numeric_cols] = scaler.fit_transform(data_cleaned[numeric_cols])
```

```
# 将SeriousDlqin2yrs设为目标变量
target_variable = 'SeriousDlqin2yrs'

from sklearn.model_selection import train_test_split

# 定义特征和目标
X = data_cleaned.drop(columns=[target_variable])
y = data_cleaned[target_variable]

# 划分数据
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 显示划分后的数据形状
print(f'训练数据形状: {X_train.shape}')
print(f'测试数据形状: {X_test.shape}')

# 保存清洗后的数据到CSV
cleaned_file_path = '2.1.3_cleaned_data.csv'
data_cleaned.to_csv(cleaned_file_path, index=False)
```

2.1.4 ipynb



```python
import pandas as pd

# 加载数据集并指定编码
file_path = 'medical_data.csv'
data = pd.read_csv(file_path, encoding='gbk')

# 查看数据类型
print(data.dtypes)
# 查看表结构
print(data.info())

# 显示每一列的空缺值数量
print(data.isnull().sum())

# 规范日期格式
data['就诊日期'] = pd.to_datetime(data['就诊日期'])
data['诊断日期'] = pd.to_datetime(data['诊断日期'])

# 重命名列
data.rename(columns={'病人ID': '患者ID'}, inplace=True)

# 查看修改后的表结构
print(data.head())
```



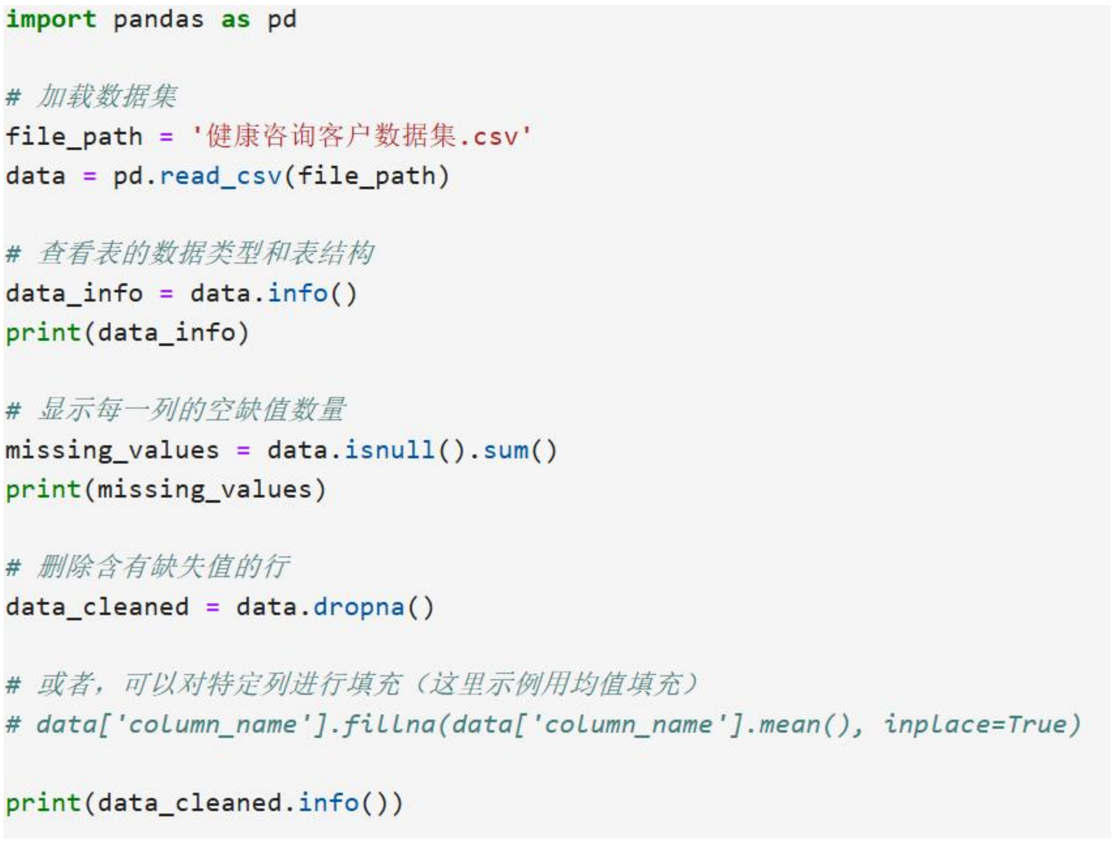
```python
from datetime import datetime

# 增加诊断延迟和病程
data['诊断延迟'] = (data['诊断日期'] - data['就诊日期']).dt.days
data['病程'] = (datetime(2024, 9, 1) - data['诊断日期']).dt.days

# 删除不合理的数据
data = data[(data['诊断延迟'] >= 0) & (data['年龄'] > 0) & (data['年龄'] < 120)]

# 查看修改后的数据
print(data.describe())

# 删除重复值并记录删除的行数
initial_rows = data.shape[0]
data.drop_duplicates(inplace=True)
deleted_rows = initial_rows - data.shape[0]

print(f'删除的重复行数: {deleted_rows}')

from sklearn.preprocessing import MinMaxScaler
```

```python
# 对需要归一化的列进行处理
scaler = MinMaxScaler()
columns_to_normalize = ['年龄', '体重', '身高']
data[columns_to_normalize] = scaler.fit_transform(data[columns_to_normalize])

# 查看归一化后的数据
print(data.head())

import matplotlib.pyplot as plt
import matplotlib.font_manager as fm


# 统计治疗结果分布
treatment_outcome_distribution = data.groupby('疾病类型')['治疗结果'].value_counts().unstack()

# 设置中文字体
font_path = 'C:/Windows/Fonts/simhei.ttf'  # 根据你的系统调整字体路径
my_font = fm.FontProperties(fname=font_path)
```

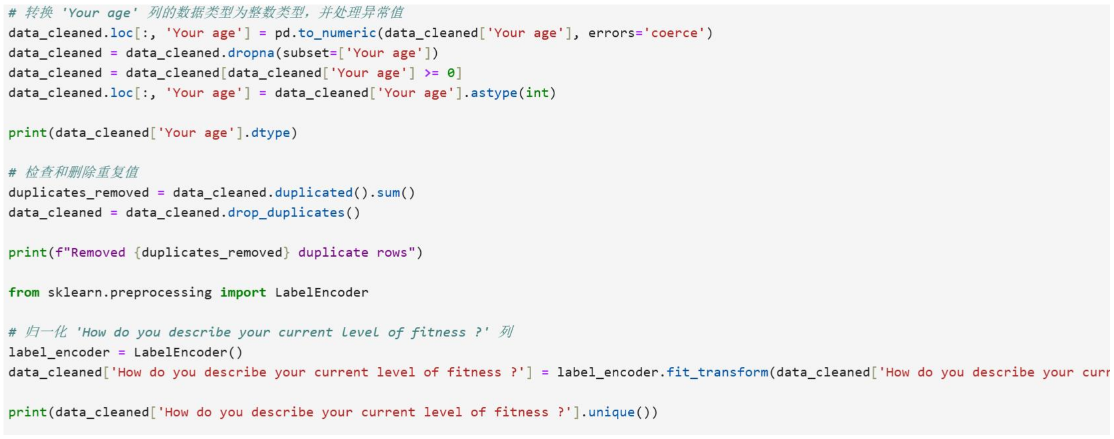
```python
# 绘制柱状图
treatment_outcome_distribution.plot(kind='bar', stacked=True)
plt.title('不同疾病类型的治疗结果分布', fontproperties=my_font)
plt.xlabel('疾病类型', fontproperties=my_font)
plt.ylabel('治疗结果数量', fontproperties=my_font)
plt.xticks(fontproperties=my_font)  # 设置x轴刻度标签的字体
plt.yticks(fontproperties=my_font)  # 设置y轴刻度标签的字体
plt.legend(prop=my_font)  # 设置图例字体
plt.show()

# 绘制散点图
plt.scatter(data['年龄'], data['疾病严重程度'])
plt.title('年龄和疾病严重程度的关系', fontproperties=my_font)
plt.xlabel('年龄', fontproperties=my_font)
plt.ylabel('疾病严重程度', fontproperties=my_font)
plt.xticks(fontproperties=my_font)  # 设置x轴刻度标签的字体
plt.yticks(fontproperties=my_font)  # 设置y轴刻度标签的字体
plt.legend(prop=my_font)  # 设置图例字体
plt.show()
```

```python
# 保存数据
output_path = '2.1.4_cleaned_data.csv'
data.to_csv(output_path, index=False)
```

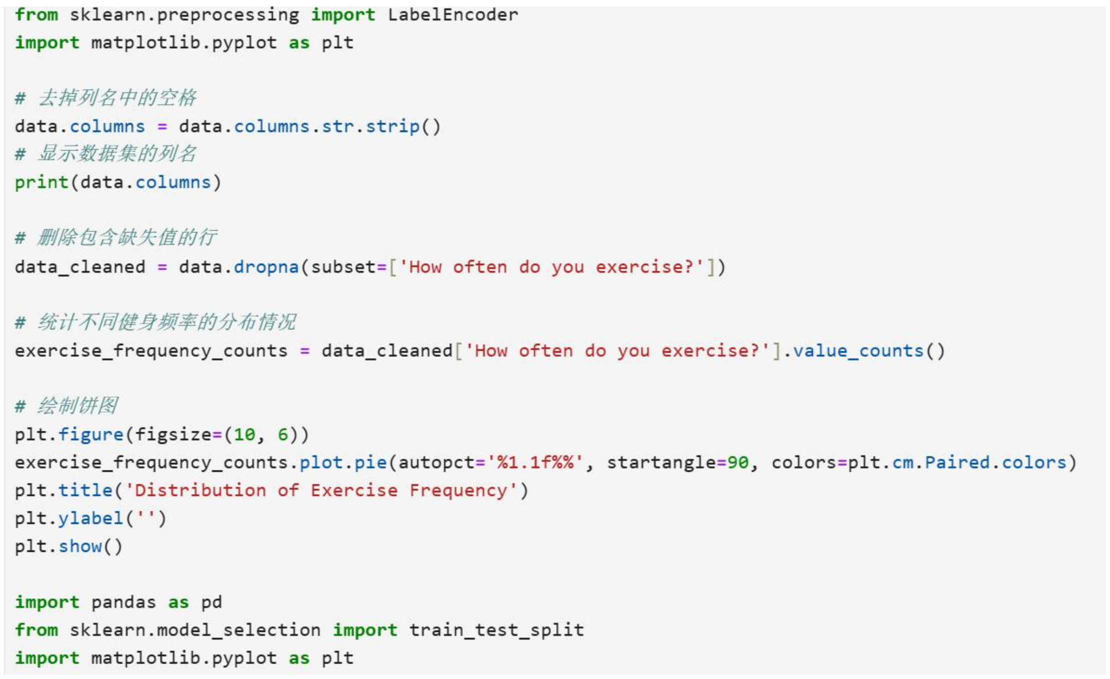
2.1.5 ipynb



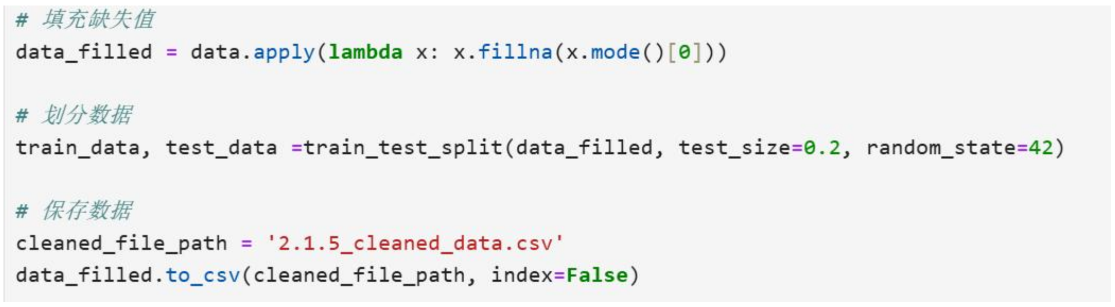
```python
import pandas as pd

# 加载数据集
file_path = '健康咨询客户数据集.csv'
data = pd.read_csv(file_path)

# 查看表的数据类型和表结构
data_info = data.info()
print(data_info)

# 显示每一列的空缺值数量
missing_values = data.isnull().sum()
print(missing_values)

# 删除含有缺失值的行
data_cleaned = data.dropna()

# 或者，可以对特定列进行填充（这里示例用均值填充）
# data['column_name'].fillna(data['column_name'].mean(), inplace=True)

print(data_cleaned.info())
```



```python
# 转换 'Your age' 列的数据类型为整数类型，并处理异常值
data_cleaned.loc[:, 'Your age'] = pd.to_numeric(data_cleaned['Your age'], errors='coerce')
data_cleaned = data_cleaned.dropna(subset=['Your age'])
data_cleaned = data_cleaned[data_cleaned['Your age'] >= 0]
data_cleaned.loc[:, 'Your age'] = data_cleaned['Your age'].astype(int)

print(data_cleaned['Your age'].dtype)

# 检查和删除重复值
duplicates_removed = data_cleaned.duplicated().sum()
data_cleaned = data_cleaned.drop_duplicates()

print(f"Removed {duplicates_removed} duplicate rows")

from sklearn.preprocessing import LabelEncoder

# 归一化 'How do you describe your current level of fitness ?' 列
label_encoder = LabelEncoder()
data_cleaned['How do you describe your current level of fitness ?'] = label_encoder.fit_transform(data_cleaned['How do you describe your curr

print(data_cleaned['How do you describe your current level of fitness ?'].unique())
```



```python
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

# 去掉列名中的空格
data.columns = data.columns.str.strip()
# 显示数据集的列名
print(data.columns)

# 删除包含缺失值的行
data_cleaned = data.dropna(subset=['How often do you exercise?'])

# 统计不同健身频率的分布情况
exercise_frequency_counts = data_cleaned['How often do you exercise?'].value_counts()

# 绘制饼图
plt.figure(figsize=(10, 6))
exercise_frequency_counts.plot.pie(autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
plt.title('Distribution of Exercise Frequency')
plt.ylabel('')
plt.show()

import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```



```python
# 填充缺失值
data_filled = data.apply(lambda x: x.fillna(x.mode()[0]))

# 划分数据
train_data, test_data =train_test_split(data_filled, test_size=0.2, random_state=42)

# 保存数据
cleaned_file_path = '2.1.5_cleaned_data.csv'
data_filled.to_csv(cleaned_file_path, index=False)
```