第一部分　上 机 实 验

实验一

Python数据分析基础实验

一、实验目的

本实验要求学生了解程序的流程控制与实现；掌握如何正确地设定循环条件，以及如何控制循环的次数；掌握函数的定义和调用方法；掌握文件操作的程序设计方法。

二、实验内容

(1) 程序流程控制实验。
(2) 函数定义及调用实验。
(3) 文件读/写操作实验。

三、实验指导

1. 程序流程控制

实验 1.1 输入一串字符，输出其中字母、数字和其他字符的个数。

程序代码如下：

```
str_s=input('输入一串字符:')
s_zm=0
s_sz=0
s_qt=0
for i in range(0,len(str_s)):
    if 'a'<=str_s[i]<='z' or 'A'<=str_s[i]<='Z':
        s_zm=s_zm+1
    elif '0'<=str_s[i]<='9':
        s_sz=s_sz+1
    else:
        s_qt=s_qt+1
```

```
print('字母字符个数:',s_zm,'数字字符个数:',s_sz,'其他字符个数:',s_qt)
```

【分析讨论】

(1) 该程序功能是否能用两路分支语句完成？试验证之。

(2) 如何使程序中的每个处理语句都执行一次？为了找出程序中每个处理语句中的错误，应该使用什么样的数据对程序进行测试？请上机验证自己的结论。

实验 1.2 计算 e 的近似值(使误差小于给定的 detax)。

程序代码如下：

```
detax=0.000001
e=1.0
x=1.0
y=1/x
i=1
while y>=detax:
    x=x*i
    y=1/x
    e=e+y
    i=i+1
print('e=',e)
```

【分析讨论】

(1) 阅读上面的程序，写出程序所依据的计算公式。

(2) 当输入的 detax 是什么值时能使程序按下面的要求运行：①不进入循环；②只循环一次；③只循环两次；④进入死循环(程序将永远循环下去)。

(3) 为了能知道程序循环了多少次，应该在程序中增加一条什么样的语句？

2. 函数定义及调用

实验 1.3 定义每月有多少天的函数，输入某年某月某日，调用该函数输出这一天是该年的第几天。

程序代码如下：

```
def days(year,month):
    if month in [1,3,5,7,8,10,12]:
        day=31
    if month in [2,4,6,9,11]:
        day=30
    if month==2:
        if (year%400==0)or(year%100!=0 and year%4==0):
            day=29
        else:
            day=28
    return day
y=int(input('输入年份:'))
m=int(input('输入月份(1~12):'))
d=int(input('输入日数(1~31):'))
day_sum=d
```

```
for i in range(1,m):
    day_sum=day_sum+days(y,i)
    print(i)
ts=str(y)+'年'+str(m)+'月'+str(d)+'日是'+str(y)+'年的第'+str(day_sum)+'天'
print(ts)
```

【分析讨论】

(1) 利用输入的一组具体数据分析程序的运行流程。

(2) 如果限制月份为1～12、日数为1～31,输入提示错误,程序应该如何修改?

(3) 通过该实验体会函数调用的优势。

3. 文件读/写操作

实验 1.4　文件读/写操作。

(1) 用键盘输入浮点数字符串(每个浮点数占一行),以'?'结束,并将数据存储到D盘data_mining_sy文件夹下的test1.txt文件中。

程序代码如下:

```
f=open('D:/data_mining_sy/test1.txt','a+')
f.truncate(0)                              #清空文件内容
data=input('输入浮点数字符串(以?结束):')
while data!='?':
    data_str=data+'\n'
    f.write(data_str)
    data=input('输入浮点数字符串(以?结束):')
f.close()
```

(2) 逐行读出字符,转换为浮点数进行相加并输出和。

程序代码如下:

```
fp=open('D:/data_mining_sy/test1.txt')
s=0
while True:
    line=fp.readline()
    tr=line[0:len(line)]
    if line!='':
      s=s+float(tr)
    else:
      break
print('s=',s)
```

【分析讨论】

(1) 在向test1.txt文件中写内容时,如果没有语句f.truncate(0),会出现什么情况?

(2) 在向test1.txt文件中写内容时,如果将语句data_str=data+'\n'换成data_str=data,会出现什么情况?

(3) 在逐行读出文本数据时,语句tr=line[0:len(line)]是必要的吗?是否可以去掉?

四、注意事项

(1) 在流程控制语句中条件的边界要清晰,实验数据要准确地覆盖各个分支。

(2) 函数定义的范围要适宜,模块不要太大。

(3) 在写入和读出文件时要注意文件的打开方式。

五、思考题

(1) 列表(List)、元组(Tuple)、集合(Set)、字典(Dictionary)是在数据挖掘实验中经常用到的,它们在应用上有何不同?

(2) open()方法用于打开一个文件,创建一个 file 对象,可以使用哪些方法调用它进行读/写?

(3) 在读/写文件时有一个文件指针记录读取的位置,数据读到哪个位置,这个指针就指到哪个位置,继续读取数据会从该位置继续读取,这样理解是否正确?

实验二

Python常用库函数应用实验

一、实验目的

本实验要求学生掌握 Pandas 的 DataFrame 数据表的创建、查询、分组和聚合等操作；掌握使用 Matplotlib 的 plot()、pie()等函数进行绘图；了解 Scikit-learn 自带的小规模数据集 iris、boston 和 digits 的数据结构，掌握 Scikit-learn 生成指定模式和复杂形状数据样本集的方法。

二、实验内容

(1) DataFrame 数据表操作实验。

(2) Matplotlib 基础绘图实验。

(3) Scikit-learn 数据集实验。

三、实验指导

1. DataFrame 数据表

实验 2.1 数据表的基本操作。

程序代码如下：

```
import pandas as pd
datas=[['S1','许文秀','女',21,'计算机系'],
       ['S2','于金凤','女',20,'计算机系'],
       ['S3','刘世元','男',22,'电信系'],
       ['S4','周新娥','女',20,'管理系'],
       ['S5','刘德峰','男',22,'电信系'],
       ['S6','吕占英','女',21,'管理系']]
column_index=['学号','姓名','性别','年龄','系部']
df=pd.DataFrame(datas,columns=column_index)
print(df)
```

```
print('\n',df.iloc[[1,3,5],[1,2,4]])                                    #①
print('\n',df[(df['性别']=='女')&(df['系部']=='计算机系')])              #②
df['籍贯']=['河北省','天津市','河北省','重庆市','江苏省','天津市']  #③
print('\n',df)
df1=df.drop([2],axis=0,inplace=False)                                   #④
print('\n',df1)
df1.drop('系部',axis=1,inplace=True)                                    #⑤
print('\n',df1)
df1.rename(columns={'学号':'sno','姓名':'name','性别':'sex','年龄':'age','籍
贯':'birthplace'},inplace=True)                                         #⑥
print('\n',df1)
```

【分析讨论】

(1) 说明注释①～⑥完成的功能。

(2) 编程修改学生“吕占英”的系部为“电信系”。

(3) 编程查询天津市学生的所有信息。

实验 2.2 数据表的分组和聚合。

程序代码如下：

```
import pandas as pd
datas=[['S1','C1',78],['S1','C2',82],['S1','C3',92],['S2','C1',67],['S2','C2',
80],['S3','C1',54],['S3','C2',68],['S3','C3',78],['S4','C1',68],['S4','C2',
84],['S4','C3',74],['S5','C2',80],['S5','C3',90],['S6','C2',80]]
column_index=['学号','课程号','成绩']
df=pd.DataFrame(datas,columns=column_index)
print('df=\n',df)
group_sno=df['成绩'].groupby(df['学号'])
group_cno=df['成绩'].groupby(df['课程号'])
print('\n',group_sno.sum())                                             #①
print('\n',group_cno.mean())                                            #②
```

【分析讨论】

(1) 说明注释①、②完成的功能。

(2) 修改程序，输出每个学生的平均成绩。

(3) 修改程序，将每个学生的总成绩和每门课的平均成绩分别存储到列表 lst_sum 和 lst_mean 中并输出。

2. Matplotlib 基础绘图

实验 2.3 使用 plot()函数绘图。

程序代码如下：

```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0.05,10,1000)
y=np.cos(x)
plt.rcParams['font.family']='STSong'          #图形中显示的汉字的字体
plt.rcParams['font.size']=12                  #显示的汉字的大小
plt.plot(x,y,ls='-',lw=2,label='函数 y=cos(x)')
```

```
plt.legend()                                    #①
plt.show()
```

【分析讨论】

(1) 如果去掉注释①语句会出现什么结果？说明该语句的作用。

(2) 修改上述程序，绘制[0，2π]区间的 y＝cos(x)的图像。

(3) 绘制[0，2π]区间的 y＝sin(x)的图像。

实验 2.4　使用 text()函数绘图。

程序代码如下：

```
port matplotlib.pyplot as plt
import numpy as np
x_axis_data=['语文','数学','英语','Python','数据库技术','Java','数据挖掘']
y_axis_data1=[68,82,69,69,70,72,66]
y_axis_data2=[71,73,52,66,74,82,71]
y_axis_data3=[82,83,82,76,84,92,81]
plt.rcParams['font.family']='STSong'
plt.rcParams['font.size']=12
plt.plot(x_axis_data,y_axis_data1,'b*--',alpha=0.5,linewidth=1,label='许文秀')#'
plt.plot(x_axis_data,y_axis_data2,'rs--',alpha=0.5,linewidth=1,label='周新娥')
plt.plot(x_axis_data,y_axis_data3,'go--',alpha=0.5,linewidth=1,label='刘世元')
for a,b in zip(x_axis_data,y_axis_data1):
    plt.text(a,b,str(b),ha='center',va='bottom',fontsize=8)
for a,b1 in zip(x_axis_data,y_axis_data2):
    plt.text(a,b1,str(b1),ha='center',va='bottom',fontsize=8)
for a,b2 in zip(x_axis_data,y_axis_data3):
    plt.text(a,b2,str(b2),ha='center',va='bottom',fontsize=8)
plt.legend()
plt.xlabel('考试科目')
plt.ylabel('考试成绩')
plt.show()
```

【分析讨论】

(1) 根据程序运行结果说明程序实现的功能。

(2) 根据程序运行结果并结合相关资料说明 zip()函数的应用及各参数的功能。

(3) 根据程序运行结果并结合主教材中的例 3.61 说明 text()函数的应用及各参数的功能。

实验 2.5　绘制饼图。

程序代码如下：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#解决中文乱码问题
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['font.size']=12
df=pd.read_excel('D://data_mining_sy/拼多多平台子类目销售额占比.xlsx')
```

```
plt.figure(figsize=(10,6))
x=df['销售额(亿元)']
labels=df['子类目']
explode=[0.1,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05]
plt.pie(x, labels = labels, autopct = '%3.1f%%', labeldistance = 1.02, explode =
explode,shadow=True)
plt.show()
```

【分析讨论】

(1) 说明 pie()函数中各参数的功能(教材上没有讲到的参数可查找相关资料)。

(2) 将参数 explode 调整为[0.3,0.1,0.1,0.05,0.05,0.05,0.05,0.05,0.05]后查看结果,并说明该参数的作用。

(3) 说明参数 shadow 的作用,并修改程序验证。

3. Scikit-learn 数据集

实验 2.6 查看 Scikit-learn 自带的小规模数据集的数据结构。

程序代码如下:

```
from sklearn.datasets import load_iris
from sklearn.datasets import load_boston
from sklearn.datasets import load_digits
data=load_iris()
print('*'*30,'(一)','*'*30)
print(data.feature_names)
print('-'*60)
print(data.target_names)
print('-'*60)
print(data.target)
print('*'*30,'(二)','*'*30)
boston=load_boston()
print(boston.data.shape)
print('-'*60)
print(boston.feature_names)
print('*'*30,'(三)','*'*30)
digit=load_digits(n_class=5,return_X_y=False)
print(digit.feature_names)
print('-'*60)
print(digit.target_names)
print('*'*60)
```

【分析讨论】

(1) 利用程序运行结果分析 Scikit-learn 自带的小规模数据集的数据结构。

(2) 修改程序,查看 Scikit-learn 自带的小规模数据集 iris、boston 和 digits 各包含多少条记录,各有多少类别。

实验 2.7 生成分类数据和聚类数据。

(1) 生成分类数据。

程序代码如下:

```
from sklearn.datasets import make_classification
X,y=make_classification(n_samples=10000,n_features=25,n_informative=3,
n_redundant=2,n_classes=3,n_clusters_per_class=1)
print(X.shape)
```

【分析讨论】

① 运行程序，并分析程序的运行结果。

② 取生成数据集的前两个特征作为二维空间中的横坐标和纵坐标，绘制出散点图。

③ 修改程序，生成二维分类数据集，并绘制出散点图。

(2) 生成聚类数据。

程序代码如下：

```
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
X,y=make_blobs(n_samples=500,n_features=2,centers=4,random_state=1)
fig,ax1=plt.subplots(1)
ax1.scatter(X[:,0],X[:,1],marker='o',s=8)
plt.show()
```

【分析讨论】

① 运行程序，并分析程序的运行结果。

② 说明参数 centers 的作用，将 centers 的值分别改为 3 和 5 后运行程序，并分析结果。

(3) 生成环形数据。

程序代码如下：

```
from sklearn.datasets import make_circles
import matplotlib.pyplot as plt
x1,y1=make_circles(n_samples=400,factor=0.5,noise=0.1)
plt.scatter(x1[:,0],x1[:,1],marker='o',c=y1,s=10,cmap='viridis')
plt.show()
```

【分析讨论】

① 运行程序，并分析程序的运行结果。

② 说明参数 noise 的作用，将 noise 的值分别改为 0.5 和 0.05 后运行程序，并分析结果。

(4) 生成月亮形数据。

程序代码如下：

```
from sklearn.datasets import make_moons
import matplotlib.pyplot as plt
import numpy as np
X,y=make_moons(n_samples=1000,noise=0.1)
plt.scatter(X[:,0],X[:,1],marker='o',c=y)
plt.show()
```

【分析讨论】

① 运行程序，并分析程序的运行结果。

② 说明参数 noise 的作用，将 noise 的值分别改为 0.5 和 0.05 后运行程序，并分析结果。

(5) 生成多维正态分布数据。

程序代码如下：

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_gaussian_quantiles
X,y=make_gaussian_quantiles(n_samples=1000,n_features=2,n_classes=3,mean=
[1,2],cov=2)
plt.scatter(X[:,0],X[:,1],marker='o',c=y)
plt.show()
```

【分析讨论】

① 运行程序，并分析程序的运行结果。

② 说明参数 mean 和 cov 的作用，修改这两个参数的值运行程序，分析其对所产生数据分布的影响。

四、注意事项

(1) iloc[]和 loc[]的区别。

(2) 指定图例大小、位置、标签图形的 legend()函数的应用。

(3) 不同的数据集是根据不同的需求生成的。

五、思考题

(1) 在 DataFrame 数据集上如何实现 SQL 的多表连接查询功能？

(2) 通过以上实验的可视化结果思考噪声对数据分布的影响。

实验三

数据相似性与可视化实验

一、实验目的

本实验要求学生了解向量间距离的概念，熟练掌握各种距离的计算；掌握散点图的绘制方法；了解词云图的作用，掌握词云图的绘制方法。

二、实验内容

(1) 向量间距离与相似性度量实验。
(2) 绘制展示数据分布的散点图实验。
(3) 创建文本词云图实验。

三、实验指导

1. 向量间距离与相似性度量

实验 3.1 假设向量 x、y 由列表 a=[1,2,3,4]和 b=[3,3,1,4]生成，完善程序测试各距离函数的值。

程序代码如下：

```
def EucliDistance(x,y):                                    #欧几里得距离
    return np.sqrt(np.sum(np.square(x-y)))
def ManhatDistance(x,y):                                   #曼哈顿距离
    return np.sum(np.abs(x-y))
def ChebyDistance(x,y):
    return np.max(np.abs(x-y))                             #切比雪夫距离
def MinKowDistance(x,y,p):
    return np.power(np.sum(np.power(np.abs(x-y),p)),1/p)   #闵可夫斯基距离
def Lp(x,p):                                               #L(p)范数
    return np.power(np.sum(np.power(np.abs(x),p)),1/p)
```

```
def CosDistance(x,y):                                    #余弦距离
    return np.inner(x,y)/np.sqrt(np.inner(x,x) * np.inner(y,y))
def CorDistance(x,y):                                    #相关距离(皮尔逊相关系数)
    return 1-np.corrcoef(x,y)[0,1]
def JaccardDistance(x,y):                                #杰卡德距离
    return 1-len(np.intersect1d(x,y))/len(np.union1d(x,y))
```

【分析讨论】

(1) 分析程序的运行结果,理解各距离的含义。

(2) 对于闵可夫斯基距离,当 p=1 时为曼哈顿距离;当 p=2 时为欧氏距离;当 $p\to\infty$ 时为切比雪夫距离。编写程序并运行,利用结果验证该结论。

2. 绘制展示数据分布的散点图

实验 3.2 将鸢尾花数据集(iris)中的 data 数据进行切片,只取前两列,利用 Python 绘制二维散点图。

程序代码如下:

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
iris=load_iris()
y=iris.target
X1=iris.data[:, :2]
plt.rcParams['font.family']='STSong'
plt.rcParams['font.size']=12
plt.scatter(X1[y==0, 0],X1[y==0, 1],color='r',marker='+')
plt.scatter(X1[y==1, 0],X1[y==1, 1],color='g',marker='x')
plt.scatter(X1[y==2, 0],X1[y==2, 1],color='b',marker='o')
plt.xlabel('sepal width')
plt.ylabel('sepal length')
plt.title('sepal 散点图')
plt.show()
```

【分析讨论】

(1) 分析程序的运行结果,说明该程序完成的功能。

(2) 将鸢尾花数据集中的 data 数据进行切片,只取后两列绘制二维散点图。

(3) 修改程序,只绘制类别为 1 的鸢尾花散点图。

实验 3.3 三维散点图。

程序代码如下:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
x=np.random.uniform(0,1,200)
y=np.random.uniform(0,1,200)
z=np.random.uniform(0,1,200)
color=np.random.uniform(0,1,200)                         #①
```

```
ax=plt.subplot(111, projection='3d')
ax.scatter(x, y, z, c=color)
ax.set_zlabel('Z')
ax.set_ylabel('Y')
ax.set_xlabel('X')
plt.show()
```

【分析讨论】

(1) 分析程序的运行结果,说明该程序完成的功能。

(2) 分别说明注释①所在行语句和之前 3 个 np.random.uniform()语句的功能。

(3) 参照该实验绘制数据集 data 为 np.array([(1, 8, 7), (2, 8, 8), (5, 1, 2), (4, 1, 1), (3, 1, 8),(2,4,9),(8,7,6),(5,4,8)]).T 的三维散点图。

实验 3.4　绘制散点图矩阵。

程序代码如下:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
v1=np.random.normal(0, 1, 100)
v2=np.random.randint(0, 23, 100)
v3=v1 * v2
df=pd.DataFrame([v1, v2, v3]).T
pd.plotting.scatter_matrix(df, diagonal='kde', color='b')
plt.show()
```

【分析讨论】

(1) 运行程序,分析程序的运行结果。

提示:运行结果图分为对角线部分和非对角线部分。其中,对角线部分是核密度估计图(Kernel Density Estimation),用来查看某个变量的分布情况,横轴对应该变量的值,纵轴对应该变量的密度(可以理解为出现的频率);非对角线部分是两个变量之间分布的关联散点图,将任意两个变量进行配对,以其中一个为横坐标,另一个为纵坐标,将所有的数据点绘制在图上,用来衡量两个变量的关联度(Correlation)。

(2) 将程序中的 v1 改为 np.array([1,2,3,4,5,6,7,8])、v2 改为 np.array([2,8,6,4,3,6,8,3]),根据具体的数据分析结果。

3. 创建文本词云图

实验 3.5　根据 D 盘 data_mining_sy 文件夹下 cyt.txt 文件中的文本创建词云图。

(1) 直接将文本用词云图展现。

程序代码如下:

```
from wordcloud import WordCloud
with open("D:/data_mining_sy/cyt.txt",encoding="utf-8")as file:
text=file.read()
    wordcloud=WordCloud(font_path="C:/Windows/Fonts/simfang.ttf",background_
```

```
color="black",width=600,height=300,max_words=50).generate(text)
    image=wordcloud.to_image()
    image.show()
```

(2) 利用 jieba 分词工具展现词云图。

程序代码如下：

```
from wordcloud import WordCloud
import jieba
f=open("D:/data_mining_sy/cyt.txt",encoding="utf-8")
text=f.read()
t=jieba.lcut(text)
tt=' '.join(t)
wordcloud= WordCloud(font_path="C:/Windows/Fonts/simfang.ttf", background_
color="black", width=600,height=300, max_words=50,min_font_size=8).generate
(tt)
image=wordcloud.to_image()
image.show()
```

(3) 将词云图展现为指定形状。

程序代码如下：

```
from wordcloud import WordCloud
import numpy as np
import jieba
import PIL.Image as Image
f=open("D:/data_mining_sy/cyt.txt",encoding="utf-8")
text=f.read()
t=jieba.lcut(text)
tt=' '.join(t)
mask_pic=np.array(Image.open('D:/data_mining_sy/14.jpg'))
wordcloud= WordCloud(font_path="C:/Windows/Fonts/simfang.ttf", background_
color='white',mask=mask_pic).generate(tt)
image=wordcloud.to_image()
image.show()
```

【分析讨论】

(1) 运行这 3 个词云图创建程序，分析、比较运行结果。

(2) 检索相关资料，掌握 jieba 的 lcut()函数对中文文本进行词切分的方法，体会使用 join()函数对词进行连接的意义。

(3) 分析 WordCloud()中参数 mask 的应用。

四、注意事项

(1) 向量距离的大小是很多算法中的重要参考数据。

(2) 散点图常用来显示数据分布，比较几个变量的相关性或者分组。

(3) 词云图过滤掉大量低频、低质的文本信息，浏览者只要一眼扫过词云图就可以领

会整篇文档的主旨。

（4）WordCloud()中参数 mask 所应用的图形一般是无背景图片。

五、思考题

（1）向量之间的距离与它们之间的相似度有什么关系？

（2）为什么说散点图是用来判断两个变量之间相互关系的工具？

（3）词云图有哪些优点和缺点？

实验四

数据采集与预处理实验

一、实验目的

本实验要求学生了解数据采集的方法；能够利用 Pandas 进行缺失值处理和异常值处理；掌握 DataFrame 数据表连接的方法；能够利用主成分分析进行数据归约；掌握连续数据离散化的方法。

二、实验内容

(1) 数据采集实验。
(2) 缺失值处理实验。
(3) 异常值处理实验。
(4) 数据表连接实验。
(5) 数据归约实验。
(6) 数据离散化实验。

三、实验指导

1. 数据采集

urllib 库是 Python 内置的 HTTP 请求库，它包含 4 个模块，第一个模块 request 是最基本的 HTTP 请求模块；第二个模块 error 是异常处理模块；第三个模块 parse 是一个工具模块，提供了许多 URL 处理方法；第四个模块 robotparser 主要用来识别网站的 robots.txt 文件。

urlopen()方法可以实现请求的发起，如果要加入 Headers 等信息，可以使用 Request 类来构造请求。其使用方法如下：

```
urllib.request.Request(url, data=None, headers={}, origin_req_host=None,
unverifiable=False,method=None)
```

参数说明：

(1) url 为要请求的 URL。

(2) data 必须是 bytes(字节流)类型。

(3) headers 是一个字典类型，为请求头部，可以在构造请求时通过 headers 参数直接构造，也可以通过调用请求实例的 add_header()方法添加。

(4) origin_req_host 指定请求方的 Host 名称或者 IP 地址。

(5) unverifiable 设置网页是否需要验证，默认为 False，该参数一般不用设置。

(6) method 是一个字符串，用来指定请求使用的方法，例如 GET、POST 和 PUT 等。

实验 4.1　在“证券之星”网站上获取某网页中的 A 股数据。

该实验主要分为三部分，即网页源代码的获取、所需内容的提取、所得结果的整理。

程序代码如下：

```
import urllib.request
import re
import csv
import os
url='http://quote.stockstar.com/stock/ranklist_a_3_1_1.html' #目标网址
headers={"User-Agent":"Mozilla/5.0 (Windows NT 10.0; WOW64)"} #伪装浏览器请求报头
request=urllib.request.Request(url=url,headers=headers)  #请求服务器
response=urllib.request.urlopen(request)        #服务器应答
content=response.read().decode('gbk')           #以一定的编码方式查看源代码
pattern=re.compile('<tbody[\s\S]*</tbody>')
body=re.findall(pattern,str(content))    #匹配<tbody>和</tbody>之间的所有代码
pattern=re.compile('>(.*?)<')
stock_page=re.findall(pattern,body[0])          #匹配>和<之间的所有信息
stock_total=stock_page
stock_last=stock_total[:]                       #stock_total:匹配出的股票数据
for data in stock_total:                        #stock_last:整理后的股票数据
    if data=='':
        stock_last.remove('')
head=['代码','简称','最新价','涨跌幅','涨跌额','5分钟涨幅']
lst=[]
for i in range(0,len(stock_last),6):            #网页中共有6列数据
    lst.append([stock_last[i],stock_last[i+1],stock_last[i+2], stock_last[i
+3],stock_last[i+4],stock_last[i+5]])
    os.chdir('D:\\data_mining_sy')              #改变当前路径
    with open('股票数据.csv','a',newline='') as f: #以追加方式打开或创建文件
      f_csv=csv.writer(f)
      f_csv.writerow(head)                      #写入文件头
      for i in range(len(lst)):                 #按行写入文件
          f_csv.writerow(lst[i])
```

【分析讨论】

(1) 阅读并运行程序，了解各部分的意义。

(2) 编程读取股票数据.csv 文件。

(3) 绘制各股票“涨跌幅”折线图。

BeautifulSoup 是 Python 的一个库,其最主要的功能是从网页中抓取数据。在创建 BeautifulSoup 对象时首先要导入其对应的 bs4 库。

实验 4.2 将抓取的新浪网新闻内容存储到 D 盘 data_mining_sy 文件夹下的 titles.txt 文件中。

程序代码如下:

```
from bs4 import BeautifulSoup
import requests
from datetime import datetime
import json
import re
news_url = 'http://news.sina.com.cn/c/nd/2017-05-08/doc-ifyeycfp9368908.shtml'
web_data=requests.get(news_url)
web_data.encoding='utf-8'
soup=BeautifulSoup(web_data.text,'lxml')
title=soup.select('#artibodyTitle')[0].text
time=soup.select('.time-source')[0].contents[0].strip()
dt=datetime.strptime(time,'%Y年%m月%d日%H:%M')
source=soup.select('.time-source span span a')[0].text
editor=soup.select('.article-editor')[0].text.lstrip('责任编辑:')
comments=requests.get('http://comment5.news.sina.com.cn/page/info?version=1&format=js&channel=gn&newsid=comos-fyeycfp9368908&group=&compress=0&ie=utf-8&oe=utf-8&page=1&page_size=20')
comments_total=json.loads(comments.text.strip('var data='))
news_id=re.search('doc-i(.+).shtml',news_url)
titles=title+'\n'+str(dt)+'\n'+source+'\n'
titles=titles+str('\n'.join([p.text.strip() for p in soup.select('#artibody p')[:-1]]))
titles=titles+'\n'+editor+'\n'+str(comments_total['result'])
titles=titles+'\n'+news_id.group(1)
try:
    #以只写的方式打开或创建 titles.txt 文件
    file=open(r'D:/data_mining_sy/titles.txt', 'w')
    for title in titles:
    #将爬取到的文章题目写入文件中
        file.write(title)
finally:
    if file:
        #关闭文件(很重要)
        file.close()
```

【分析讨论】

(1) 阅读程序和注释,了解各主要语句的功能。

(2) 编程读取结果文件 titles.txt 中的内容。

(3) 考虑提取 titles.txt 文件中汉字和数字的方法。