

**广州城建职业学院**

**实验实训指导书**

**开课学院： 信息工程学院**

**课程名称：** **《Python数据分析》**

**指导老师： 于玮烨**

**所属教研室： 软件技术**

|  |  |
| --- | --- |
| **编制人** | **审核人** |
| 于玮烨 |  |

广州城建职业学院教务处

2023 年 2月

**一．实训基本信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实训类别** | 整周实训□ 理论＋实践课**√** 纯实践课□ | | | |
| **开课班级** | 22大数据技术1，2班 | | | |
| **实践学时/学分** | 32 学时/ 4 学分 | **实训项目（任务）数** | | 7 个项目  19个任务 |
| **实验实训性质** | 基础技能实训**√** 核心技能实训□ 综合技能实训□ | | | |
| **面向专业（方向）** | **软件技术** | **开设学期** | 2022-2023学年  第 2 学期 | |

**二．实验实训项目（任务）和目标设计**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **实验实训项目（任务）** | **实训场地及**  **配套设备（含软件）** | **学时数** | **实验实训目标** | **实验实训结果**  **（可检测）** | **考核标准** |
| 1 | 认识数据分析 | 2409  计算机  Python | 2 | 1．掌握数据分析的概念  2. 了解Python数据分析常用库 | 1．可以创建虚拟环境。  2．可以安装各类第三方库 | 1．创建出的虚拟环境必须满足要求的python版本。  2．可以用命令安装成功第三方库 |
| 2 | 掌握NumPy数组对象ndarray | 2409  计算机  Python | 4 | 1. 创建数组对象  2. 创建NumPy矩阵  3. 掌握ufunc函数　4. 利用NumPy进行统计分析 | 1．学生创建出数组  2．学生创建出矩阵  3．使用ufunc进行运算。  4．可以使用NumPy进行统计分析 | 1．学生创建出数组  2．学生创建出矩阵  3．使用ufunc进行运算。  4．可以使用NumPy进行统计分析 |
| 3 | pandas统计分析基础 | 2409  计算机  Python | 4 | 1. 读/写不同数据源的数据  2. 掌握DataFrame的常用操作 | 1. 成功读/写不同数据源的数据  2. 掌握DataFrame的常用操作，进行增删改查 | 1．成功读/写不同数据源的数据2．掌握DataFrame的常用操作，进行增删改查 |
| 4 | 使用pandas进行数据预处理 | 2409  计算机  Python | 4 | 1. 转换与处理时间序列数据  2. 使用groupby()方法拆分数据  3. 使用agg()方法聚合数据 | 1. 转换与处理时间序列数据，有成功转换的代码  2. 使用groupby()方法拆分数据，有成功拆分的数据结果  3. 使用agg()方法聚合数据，有聚合操作的代码和结果 | 1.有成功转换的代码  2.有成功拆分的数据结果  3.有聚合操作的代码和结果 |
| 5 | Matplotlib、seaborn、pyecharts数据可视化基础 | 2409  计算机  Python | 4 | 1. 掌握Matplotlib基础绘图  2. 掌握seaborn基础绘图  3. 掌握pyecharts基础绘图 | 1．有基于Matplotlib的绘图结果  2．有基于seaborn的绘图结果  3．有基于pyecharts的绘图结果 | 1．有基于Matplotlib的绘图结果和代码  2．有基于seaborn的绘图结果和代码  3．有基于pyecharts的绘图结果和代码 |
| 6 | 使用scikit-learn构建模型 | 2409  计算机  Python | 5 | 1. 使用scikit-learn构建模型  2. 构建并评价聚类模型  3. 构建并评价分类模型数 | 1. 有实现scikit-learn构建模型的代码  2. 有构建并评价聚类模型的代码  3. 有构建并评价分类模型数的代码 | 1．有实现各相关功能的代码  2．可以调用自己创建的函数。  3．有创建存储过程的代码，并且通过调用，看是否有结果  4．有实现创建触发器的代码 |
| 7 | **综合项目** | 2409  计算机  Python | 4 | 1. 对航空用户进行分群  2. 政府收入预测分析 | 1. 对航空用户进行分群，有完整的项目代码  2. 政府收入预测分析，有完整的项目代码 | 1. 对航空用户进行分群，有完整的项目代码  2. 政府收入预测分析，有完整的项目代码 |

目录

[任务1 Python 安装与开发环境搭建 6](#_Toc160405931)

[**一、实验实训目的** 6](#_Toc160405932)

[**二、实验实训要求** 6](#_Toc160405933)

[**三、实验实训步骤** 6](#_Toc160405934)

[**五、实验实训素材（可另附文件）** 7](#_Toc160405935)

[任务2 Python 运算符、内置函数 8](#_Toc160405936)

[任务3 使用蒙特 · 卡罗方法计算圆周率近似值 10](#_Toc160405937)

[任务4 使用列表实现筛选法求素数 12](#_Toc160405938)

[任务5 使用集合实现筛选法求素数 14](#_Toc160405939)

[任务6理解浮点数运算的误差 16](#_Toc160405940)

[任务7小明爬楼梯 18](#_Toc160405941)

[任务8聪明的尼姆游戏（人机对战） 20](#_Toc160405942)

[任务9蒙蒂霍尔悖论游戏 24](#_Toc160405943)

[**一、实验实训目的** 24](#_Toc160405944)

[**二、实验实训要求** 24](#_Toc160405945)

[**三、实验实训步骤** 24](#_Toc160405946)

[**五、实验实训素材（可另附文件）** 26](#_Toc160405947)

[任务10 猜数游戏 27](#_Toc160405948)

[**一、实验实训目的** 27](#_Toc160405949)

[**二、实验实训要求** 27](#_Toc160405950)

[**三、实验实训步骤** 27](#_Toc160405951)

[**五、实验实训素材（可另附文件）** 28](#_Toc160405952)

[任务11抓狐狸游戏 29](#_Toc160405953)

[**一、实验实训目的** 29](#_Toc160405954)

[**二、实验实训要求** 29](#_Toc160405955)

[**三、实验实训步骤** 29](#_Toc160405956)

[**五、实验实训素材（可另附文件）** 31](#_Toc160405957)

[任务12汉诺塔问题 32](#_Toc160405958)

[**一、实验实训目的** 32](#_Toc160405959)

[**二、实验实训要求** 32](#_Toc160405960)

[**三、实验实训步骤** 32](#_Toc160405961)

[**五、实验实训素材（可另附文件）** 33](#_Toc160405962)

[任务13凯撒加密 34](#_Toc160405963)

[**一、实验实训目的** 34](#_Toc160405964)

[**二、实验实训要求** 34](#_Toc160405965)

[**三、实验实训步骤** 34](#_Toc160405966)

[**五、实验实训素材（可另附文件）** 35](#_Toc160405967)

[任务14打字练习成绩评定 36](#_Toc160405968)

[**一、实验实训目的** 36](#_Toc160405969)

[**二、实验实训要求** 36](#_Toc160405970)

[**三、实验实训步骤** 36](#_Toc160405971)

[**五、实验实训素材（可另附文件）** 37](#_Toc160405972)

[任务15 批量生成随机信息 38](#_Toc160405973)

[**一、实验实训目的** 38](#_Toc160405974)

[**二、实验实训要求** 38](#_Toc160405975)

[**三、实验实训步骤** 38](#_Toc160405976)

[**五、实验实训素材（可另附文件）** 42](#_Toc160405977)

[任务16自定义类模拟三维向量及其运算 43](#_Toc160405978)

[**一、实验实训目的** 43](#_Toc160405979)

[**二、实验实训要求** 43](#_Toc160405980)

[**三、实验实训步骤** 43](#_Toc160405981)

[**五、实验实训素材（可另附文件）** 45](#_Toc160405982)

[**一、实验实训目的** 46](#_Toc160405983)

[**二、实验实训要求** 46](#_Toc160405984)

[**三、实验实训步骤** 46](#_Toc160405985)

[**五、实验实训素材（可另附文件）** 50](#_Toc160405986)

[**一、实验实训目的** 51](#_Toc160405987)

[**二、实验实训要求** 51](#_Toc160405988)

[**三、实验实训步骤** 51](#_Toc160405989)

[**五、实验实训素材（可另附文件）** 57](#_Toc160405990)

[**一、实验实训目的** 58](#_Toc160405991)

[**二、实验实训要求** 58](#_Toc160405992)

[**三、实验实训步骤** 58](#_Toc160405993)

[**五、实验实训素材（可另附文件）** 63](#_Toc160405994)

[**一、实验实训目的** 64](#_Toc160405995)

[**二、实验实训要求** 64](#_Toc160405996)

[**三、实验实训步骤** 64](#_Toc160405997)

[**五、实验实训素材（可另附文件）** 73](#_Toc160405998)

[**一、实验实训目的** 74](#_Toc160405999)

[**二、实验实训要求** 74](#_Toc160406000)

[**三、实验实训步骤** 74](#_Toc160406001)

[**五、实验实训素材（可另附文件）** 83](#_Toc160406002)

任务1 Python 安装与开发环境搭建

**一、实验实训目的**

1、熟练掌握 Python 解释器安装与基本用法。

2、熟练掌握使用 pip 命令安装 Python 扩展库。

3、熟悉离线安装轮子文件的方法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

1、安装 Python 解释器。

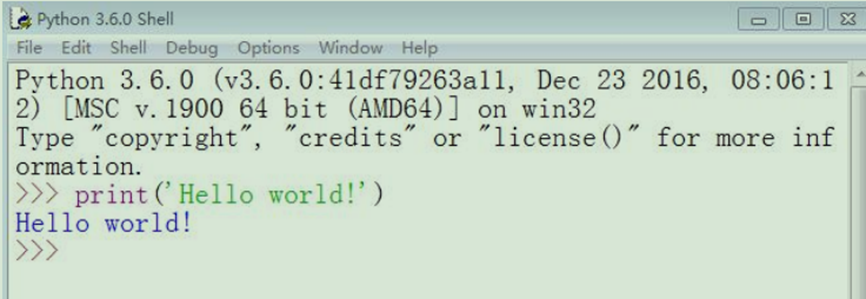
2、安装 Python 扩展库。

**实验步骤：**

1、打开 Python 官方网站 http://www.python.org。

2、下载 Python 3.5.x 或 Python 3.6.x 或 Python 3.7.x 的最新版，至少安装其中两个。

3、在开始菜单中找到成功安装的 IDLE，输入下面的代码，确保 IDLE 运行正常。

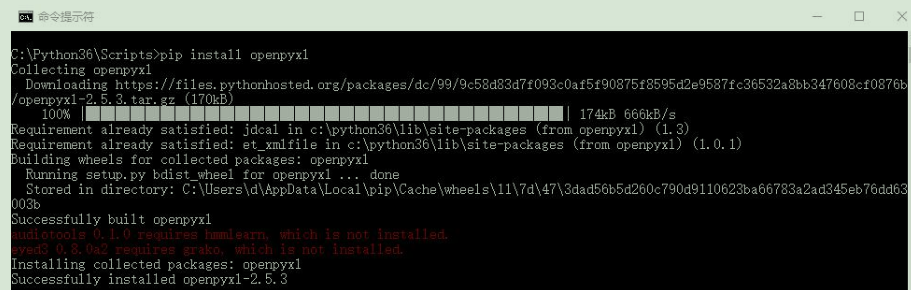
4、在资源管理器中进入 Python 安装目录的 scripts 子目录，然后按下 Shift 键，在空白

处单击鼠标右键，在弹出来的菜单中选择“在此处打开命令窗口”进入命令提示符环境。如图所示：

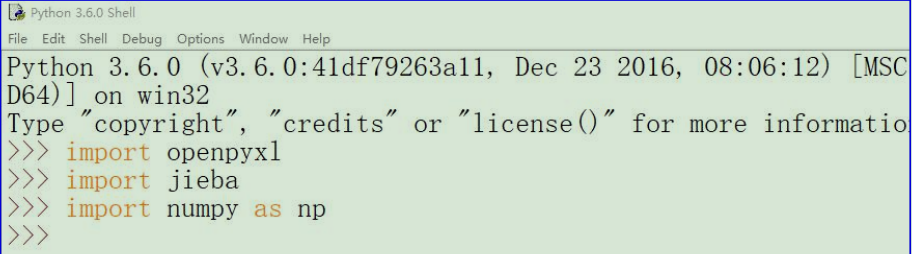
文本

低可信度描述已自动生成

5、使用 pip 命令在线安装 Python 扩展库 numpy、pandas、scipy、matplotlib、jieba、openpyxl、pillow。安装 openpyxl 的命令如图所示：

6、如果遇到安装不成功的扩展库，使用浏览器打开下面的网址下载 whl 文件进行离线安装：https://www.lfd.uci.edu/~gohlke/pythonlibs/

7、在 IDLE 中使用 import 导入安装好的扩展库，验证是否安装成功。



**四、实验实训结果与形式**

1.实现任务的相关代码

2.代码执行结果截图

**五、实验实训素材（可另附文件）**

无

任务2 Python 运算符、内置函数

**一、实验实训目的**

1、熟练运用 Python 运算符。

2、熟练运用 Python 内置函数。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

实验内容：

1、编写程序，输入任意大的自然数，输出各位数字之和。

2、编写程序，输入两个集合 setA 和 setB，分别输出它们的交集、并集和差集 setA-setB。

3、编写程序，输入一个自然数，输出它的二进制、八进制、十六进制表示形式。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

#1、

num = input('请输入一个自然数：')

print(sum(map(int, num)))

#2、

setA = eval(input('请输入一个集合：'))

setB = eval(input('再输入一个集合：'))

print('交集：', setA & setB)

print('并集：', setA | setB)

print('setA-setB：', setA - setB)

#3、

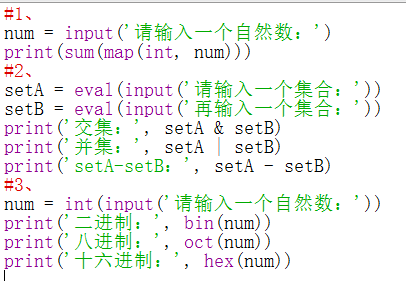
num = int(input('请输入一个自然数：'))

print('二进制：', bin(num))

print('八进制：', oct(num))

print('十六进制：', hex(num))

**2.源代码截图**



**3.运行结果截图：**

图示, 文本, 信件

描述已自动生成

**五、实验实训素材（可另附文件）**

无

任务3 使用蒙特 · 卡罗方法计算圆周率近似值

**一、实验实训目的**

1、理解蒙特·卡罗方法原理。

2、理解 for 循环本质与工作原理。

3、了解 random 模块中常用函数。

**二、实验实训要求**

操作系统：Windows 7版本及以上

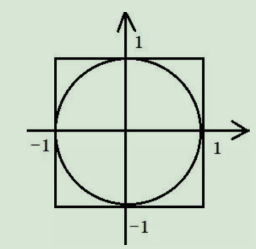
Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

蒙特·卡罗方法是一种通过概率来得到问题近似解的方法，在很多领域都有重要的应用，其中就包括圆周率近似值的计算问题。假设有一块边长为 2 的正方形木板，上面画一个单位圆，然后随意往木板上扔飞镖，落点坐标(x, y)必然在木板上（更多的时候是落在单位圆内），如果扔的次数足够多，那么落在单位圆内的次数除以总次数再乘以 4，这个数字会无限逼近圆周率的值。这就是蒙特·卡罗发明的用于计算圆周率近似值的方法，如图所示。



编写程序，模拟蒙特·卡罗计算圆周率近似值的方法，输入掷飞镖次数，然后输出圆周率近似值。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

import random

times=int(input("请输入投掷次数："))

hits=0

for i in range(times):

x=random.random()

y=random.random()

if x\*x+y\*y<=1:

hits+=1

print(4.0\*hits/times)

**2.源代码截图**

**文本

描述已自动生成**

**3.运行结果截图：**

文本

描述已自动生成

**五、实验实训素材（可另附文件）**

无

任务4 使用列表实现筛选法求素数

**一、实验实训目的**

1、理解筛选法求解素数的原理。

2、理解列表切片操作。

3、熟练运用内置函数 enumerate()。

4、熟练运用内置函数 filer()。

5、理解序列解包工作原理。

6、初步了解选择结构和循环结构。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

实验内容：

编写程序，输入一个大于 2 的自然数，然后输出小于该数字的所有素数组成的列表。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

**方法(1)**

maxNumber = int(input('请输入一个大于 2 的自然数：'))

lst = list(range(2, maxNumber))

#最大整数的平方根

m = int(maxNumber\*\*0.5)

for index, value in enumerate(lst):

#如果当前数字已大于最大整数的平方根，结束判断

if value > m:

break

#对该位置之后的元素进行过滤

lst[index+1:] = filter(lambda x: x%value != 0, lst[index+1:])

print(lst)

**方法(2)**

def Prime(n):

for i in range(2,n):

if n%i==0:

return False

return True

n=int(input('请输入一个大于 2 的自然数：\n'))

result=list()

for i in range(2,n):

if Prime(i)==True:

result.append(i)

print('这是返回的素数：',result)

**2.源代码截图**

**方法(1)**

文本

描述已自动生成

**方法(2)**

文本

描述已自动生成

**3.运行结果截图：**

图片包含 图示

描述已自动生成

**五、实验实训素材（可另附文件）**

无

任务5 使用集合实现筛选法求素数

**一、实验实训目的**

1、理解求解素数的筛选法原理。

2、理解 Python 集合对象的 discard()方法。

3、熟练运用列表推导式。

4、理解 for 循环工作原理。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

实验内容：

输入一个大于 2 的自然数，输出小于该数字的所有素数组成的集合。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

**方法(1)**

maxNumber = int(input('请输入一个大于 2 的自然数：'))

numbers = set(range(2, maxNumber))

# 最大数的平方根，以及小于该数字的所有素数

m = int(maxNumber\*\*0.5)+1

primesLessThanM = [p for p in range(2, m)

if 0 not in [p%d for d in range(2, int(p\*\*0.5)+1)]]

# 遍历最大整数平方根之内的自然数

for p in primesLessThanM:

for i in range(2, maxNumber//p+1):

# 在集合中删除该数字所有的倍数

numbers.discard(i\*p)

print(numbers)

**方法(2)**

def isprime(n):

for i in range(2, n):

if n % i == 0:

return 0

return 1

n = int(input("请输入一个数据:"))

value\_set = set()

for i in range(2, n):

if isprime(i) == 1:

value\_set.add(i)

print(value\_set)

**2.源代码截图**

**方法(1)**

手机屏幕截图

描述已自动生成

**方法(2)**

文本

描述已自动生成

**3.运行结果截图：**



**五、实验实训素材（可另附文件）**

无

任务6理解浮点数运算的误差

**一、实验实训目的**

1、理解组合数定义式的化简。

2、理解浮点数运算的误差和可能带来的问题。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

实验内容：

阅读并调试下面的代码，分析代码功能，发现并解决代码中的错误。

提示：

这段代码试图计算组合数：



但是由于浮点数除法时精度问题导致结果错误。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

**#原函数:**

def cni(n,i):

minNI = min (i,n-i)#C(8,5)->C(8,3),减少下面循环次数

result = 1

for j in range(0,minNI):

print('(',result,'\*',n -j,')/', minNI-j) #调试

result =result\*(n-j)/(minNI-j) #从最大项开始累乘

return result

print('原函数：')

print(cni(8,5),end='\n\n')

**#改正后:**

def cni1(n,i):

minNI=min(i,n-i)

result = 1

for j in range(1, minNI+1):

print('(',result,'\*',n -minNI+j,')/', j)#从最小项开始累乘

result = (result \* (n -minNI+j)) /j#为了保证能够整除，先进行乘法后除

return result

print('修改后：')

print(cni1(8,5))

**2.源代码截图**

**#原函数:**

文本

描述已自动生成

**#改正后:**

文本

描述已自动生成

**3.运行结果截图：**

文本

描述已自动生成

图片包含 文本

描述已自动生成

**五、实验实训素材（可另附文件）**

无

任务7小明爬楼梯

**一、实验实训目的**

1、理解并熟练使用序列解包。

2、理解递归函数工作原理。

3、能够编写递归函数代码解决实际问题。

4、理解 Python 字典的用法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

假设一段楼梯共 15 个台阶，小明一步最多能上 3 个台阶。编写程序计算小明上这段楼梯一共有多少种方法。要求给出递推法和递归法两种代码。

**四、实验实训结果与形式**

**1.实现任务的相关代码**

def climbStairs1(n):

#递推法

a = 1

b = 2

c = 4

for i in range(n-3):

c, b, a = a+b+c, c, b

return c

def climbStairs2(n):

#递归法

first3 = {1:1, 2:2, 3:4}

if n in first3.keys():

return first3[n]

else:

return climbStairs2(n-1) + climbStairs2(n-2) + climbStairs2(n-3)

print(climbStairs1(15))

print(climbStairs2(15))

**2.源代码截图:**

文本, 信件

描述已自动生成

**3.运行结果截图：**



**五、实验实训素材（可另附文件）**

无

任务8聪明的尼姆游戏（人机对战）

**一、实验实训目的**

1、理解尼姆游戏规则。

2、了解多个函数定义与调用。

3、理解并熟练运用 while 循环。

4、理解带 else 子句的循环结构执行流程。

5、理解循环语句中的 break 语句的作用。

6、了解使用循环和异常处理结构对用户输入进行约束的用法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

实验内容：

编写程序，模拟聪明版本的尼姆游戏。

尼姆游戏是个著名的游戏，有很多变种玩法。两个玩家轮流从一堆物品中拿走一部分。在每一步中，玩家可以自由选择拿走多少物品，但是必须至少拿走一个并且最多只能拿走一半物品，然后轮到下一个玩家。拿走最后一个物品的玩家输掉游戏。在聪明模式中，计算机每次拿走足够多的物品使得堆的大小是 2 的幂次方减 1——也就是 3,7,15,31 或 63。除了堆的大小已经是 2 的幂次方减 1，在其他情况下这样走都是符合游戏规则的。在那种情况下，计算机就按游戏规则随机拿走一些。

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**(1)首先用Python实现这个游戏：**

from random import randint

def nimu(n):

while n > 1 :

# 玩家走

print("\nNow it's your turn, and we have {0} left.".format(n))

# 确保玩家输入的是合法整数值

while True :

try:

num = int(input("How many would you to take:"))

assert 1<= num <= n//2

break

except:

print("\nThe number you can take must between 1 and {0}".format(n//2))

n -= num

if n == 1:

return "You lose!"

# 机器人随意拿走一些

n -= randint(1,n//2)

else:

return "You win!"

print(nimu(randint(1, 1000)))

**(2)利用Python来实现这个必胜机器人，代码如下：**

from math import log2, floor

from random import randint

def answer\_n(x):

return floor(log2(x/3))

def dead\_num(n):

# 定义根据任意元素数确定机器人所需要选择的数值，以使玩家必输

if log2((n+1)/3)%1 == 0:

# 无法得到使玩家必输的数值,则在规定范围内随机使用一个数值

return randint(1,n//2)

else:

# 按玩家必输模式计算出机器人应拿走的元素数量

return n-(3\*2\*\*answer\_n(n)-1)

def nimu(n):

while n > 1 :

# 玩家走

print("\nNow it's your turn, and we have {0} left.".format(n))

# 确保玩家输入的是合法整数值

while True :

try:

num = int(input("How many would you to take:"))

assert 1<= num <= n//2

break

except:

print("\nThe number you can take must between 1 and {0}".format(n//2))

n -= num

if n == 1:

return "You lose!"

# 机器人拿走使玩家进入必输模式的数值

n -= dead\_num(n)

else:

return "You win!"

print(nimu(randint(1, 1000)))

**2.源代码截图:**

**(1)首先用Python实现这个游戏：**

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**(2)利用Python来实现这个必胜机器人，代码如下：**

文本

描述已自动生成

**3.运行结果截图：**

**(1)首先用Python实现这个游戏：**

文本

描述已自动生成

**(2)利用Python来实现这个必胜机器人，代码如下：**

文本

描述已自动生成

**五、实验实训素材（可另附文件）**

无

任务9蒙蒂霍尔悖论游戏

**一、实验实训目的**

1、了解蒙蒂霍尔悖论内容。

2、了解游戏规则。

3、熟练运用字典方法和集合运算。

4、了解断言语句 assert 的用法。

5、熟练运用循环结构。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

假设你正参加一个有奖游戏节目，并且有 3 道门可选：其中一个后面是汽车，另外两个后面是山羊。你选择一个门，比如说 1 号门，主持人当然知道每个门后面是什么并且打开了另一个门，比如说 3 号门，后面是一只山羊。这时，主持人会问你"你想改选 2 号门吗？"，然后根据你的选择确定最终要打开的门，并确定你获得山羊（输）或者汽车（赢）。

编写程序，模拟上面的游戏。

**实验步骤与参考代码：**

from random import randrange

def init():

'''返回一个字典，键为 3 个门号，值为门后面的物品'''

result = {i: 'goat' for i in range(3)}

r = randrange(3)

result[r] = 'car'

return result

def startGame():

# 获取本次游戏中每个门的情况

doors = init()

# 获取玩家选择的门号

while True:

try:

firstDoorNum = int(input('Choose a door to open:'))

assert 0<= firstDoorNum <=2

break

except:

print('Door number must be between {} and {}'.format(0, 2))

# 主持人查看另外两个门后的物品情况

for door in doors.keys()-{firstDoorNum}:

# 打开其中一个后面为山羊的门

if doors[door] == 'goat':

print('"goat" behind the door', door)

# 获取第三个门号，让玩家纠结

thirdDoor = (doors.keys()-{door, firstDoorNum}).pop()

change = input('Switch to {}?(y/n)'.format(thirdDoor))

finalDoorNum = thirdDoor if change=='y' else firstDoorNum

if doors[finalDoorNum] == 'goat':

return 'I Win!'

else:

return 'You Win.'

while True:

print('='\*30)

print(startGame())

r = input('Do you want to try once more?(y/n)')

if r == 'n':

break

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务10 猜数游戏

**一、实验实训目的**

1、熟练运用选择结构与循环结构解决实际问题。

2、注意选择结构嵌套时代码的缩进与对齐。

3、理解带 else 子句的循环结构执行流程。

4、理解条件表达式 value1 if condition else value2 的用法。

5、理解使用异常处理结构约束用户输入的用法。

6、理解带 else 子句的异常处理结构的执行流程。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

编写程序模拟猜数游戏。程序运行时，系统生成一个随机数，然后提示用户进行猜测，并根据用户输入进行必要的提示（猜对了、太大了、太小了），如果猜对则提前结束程序，如果次数用完仍没有猜对，提示游戏结束并给出正确答案。

**实验步骤与参考代码：**

from random import randint

def guessNumber(maxValue=10, maxTimes=3):

# 随机生成一个整数

value = randint(1,maxValue)

for i in range(maxTimes):

prompt = 'Start to GUESS:' if i==0 else 'Guess again:'

# 使用异常处理结构，防止输入不是数字的情况

try:

x = int(input(prompt))

except:

print('Must input an integer between 1 and ', maxValue)

else:

if x == value:

# 猜对了

print('Congratulations!')

break

elif x > value:

print('Too big')

else:

print('Too little')

else:

# 次数用完还没猜对，游戏结束，提示正确答案

print('Game over. FAIL.')

print('The value is ', value)

guessNumber()

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务11抓狐狸游戏

**一、实验实训目的**

1、培养分析问题并对进行建模的能力。

2、熟练使用列表解决实际问题。

3、熟练运用选择结构和循环结构解决实际问题。

4、理解带 else 子句的循环结构执行流程。

5、理解使用异常处理结构约束用户输入的用法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

编写程序，模拟抓狐狸小游戏。假设一共有一排 5 个洞口，小狐狸最开始的时候在其中一个洞口，然后玩家随机打开一个洞口，如果里面有狐狸就抓到了。如果洞口里没有狐狸就第二天再来抓，但是第二天狐狸会在玩家来抓之前跳到隔壁洞口里。

**实验步骤与参考代码：**

from random import choice, randrange

def catchMe(n=5, maxStep=10):

'''模拟抓小狐狸，一共 n 个洞口，允许抓 maxStep 次

如果失败，小狐狸就会跳到隔壁洞口'''

# n 个洞口，有狐狸为 1，没有狐狸为 0

positions = [0] \* n

# 狐狸的随机初始位置

oldPos = randrange(0, n)

positions[oldPos] = 1

# 抓 maxStep 次

while maxStep >= 0:

maxStep -= 1

# 这个循环保证用户输入是有效洞口编号

while True:

try:

x = input('今天打算打开哪个洞口？（0-{0}）：'.format(n-1))

# 如果输入的不是数字，就会跳转到 except 部分

x = int(x)

# 如果输入的洞口有效，结束这个循环，否则就继续输入

assert 0 <= x < n, '要按套路来啊，再给你一次机会。'

break

except:

#如果输入的不是数字，就执行这里的代码

print('要按套路来啊，再给你一次机会。')

if positions[x] == 1:

print('成功，我抓到小狐狸啦。')

break

else:

print('今天又没抓到。')

# 如果这次没抓到，狐狸就跳到隔壁洞口

if oldPos == n-1:

newPos = oldPos -1

elif oldPos == 0:

newPos = oldPos + 1

else:

newPos = oldPos + choice((-1, 1))

positions[oldPos], positions[newPos] = 0, 1

oldPos = newPos

else:

print('放弃吧，你这样乱试是没有希望的。')

# 启动游戏，开始抓狐狸吧

catchMe()

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务12汉诺塔问题

**一、实验实训目的**

1、理解函数默认值参数。

2、理解函数递归。

3、熟练运行列表对象的方法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

据说古代有一个梵塔，塔内有三个底座 A、B、C，A 座上有 64 个盘子，盘子大小不等，大的在下，小的在上。有一个和尚想把这 64 个盘子从 A 座移到 C 座，但每次只能允许移动一个盘子。在移动盘子的过程中可以利用 B 座，但任何时刻 3 个座上的盘子都必须始终保持大盘在下、小盘在上的顺序。如果只有一个盘子，则不需要利用 B 座，直接将盘子从 A 移动到 C 即可。编写函数，接收一个表示盘子数量的参数和分别表示源、目标、临时底座的参数，然后输出详细移动步骤和每次移动后三个底座上的盘子分布情况。

**实验步骤与参考代码：**

def hannoi(num, src, dst, temp=None): #递归算法

if num < 1:

return

global times #声明用来记录移动次数的变量为全局变量

#递归调用函数自身，先把除最后一个盘子之外的所有盘子移动到临时柱子上

hannoi(num-1, src, temp, dst)

# 移动最后一个盘子

print('The {0} Times move:{1}==>{2}'.format(times, src, dst))

towers[dst].append(towers[src].pop())

for tower in 'ABC': #输出 3 根柱子上的盘子

print(tower, ':', towers[tower])

times += 1

#把除最后一个盘子之外的其他盘子从临时柱子上移动到目标柱子上

hannoi(num-1, temp, dst, src)

times = 1 #用来记录移动次数的变量

n = 3 #盘子数量

towers = {'A':list(range(n, 0, -1)), #初始状态，所有盘子都在 A 柱上

'B':[],

'C':[]

}

#A 表示最初放置盘子的柱子，C 是目标柱子，B 是临时柱子

hannoi(n, 'A', 'C', 'B')

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务13凯撒加密

**一、实验实训目的**

1、了解 Python 标准库 string。

2、理解凯撒加密算法原理。

3、理解切片操作。

4、熟练运用字符串对象的方法。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

编写程序，要求输入一个字符串，然后输入一个整数作为凯撒加密算法的密钥，然后输出该字符串加密后的结果。

**实验步骤与参考代码：**

import string

def kaisa(s, k):

lower = string.ascii\_lowercase #小写字母

upper = string.ascii\_uppercase #大写字母

before = string.ascii\_letters

after = lower[k:] + lower[:k] + upper[k:] + upper[:k]

table = ''.maketrans(before, after) #创建映射表

return s.translate(table)

s = input('请输入一个字符串：')

k = int(input('请输入一个整数密钥：'))

print(kaisa(s, k))

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务14打字练习成绩评定

**一、实验实训目的**

1、熟练运用内置函数 zip()、sum()、round()、isinstance()。

2、熟练运用生成器表达式。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

编写程序，模拟打字练习程序的成绩评定。假设 origin 为原始内容，userInput 为用户练习时输入的内容，要求用户输入的内容长度不能大于原始内容的长度，如果对应位置的字符一致则认为正确，否则判定输入错误。最后成绩为：正确的字符数量/原始字符串长度，按百分制输出，要求保留 2 位小数。

**实验步骤与参考代码：**

def Rate(origin, userInput):

if not (isinstance(origin, str) and isinstance(userInput, str)):

return 'The two parameters must be strings.'

if len(origin)<len(userInput):

return 'Sorry. I suppose the second parameter string is shorter.'

# 精确匹配的字符个数

right = sum(1 for oc, uc in zip(origin, userInput) if oc==uc)

return right/len(origin)

origin = '''Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.'''

userInput = '''Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

readability counts.'''

print(round(Rate(origin, userInput)\*100, 2))

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务15 批量生成随机信息

**一、实验实训目的**

1、熟练运用标准库 random 中的函数。

2、了解标准库 string 中的字符串常量。

3、理解 Python 程序中\_\_name\_\_属性的作用。

4、了解汉字编码格式。

5、熟练掌握文本文件操作方法。

6、在文件操作时养成使用上下文管理语句 with 的习惯。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

编写程序，生成 200 个人的模拟信息，包括姓名、性别、年龄、电话号码、家庭住址、电子邮箱地址，把生成的信息写入文本文件，每行存放一个人的信息，最后再读取生成的文本文件并输出其中的信息。

**实验步骤与参考代码：**

import random

import string

#常用汉字 Unicode 编码表，可以自行搜索

StringBase = '\u7684\u4e00\u4e86\u662f\u6211\u4e0d\u5728\u4eba\u4eec'\

'\u6709\u6765\u4ed6\u8fd9\u4e0a\u7740\u4e2a\u5730\u5230'\

'\u5927\u91cc\u8bf4\u5c31\u53bb\u5b50\u5f97\u4e5f\u548c'\

'\u90a3\u8981\u4e0b\u770b\u5929\u65f6\u8fc7\u51fa\u5c0f'\

'\u4e48\u8d77\u4f60\u90fd\u628a\u597d\u8fd8\u591a\u6ca1'\

'\u4e3a\u53c8\u53ef\u5bb6\u5b66\u53ea\u4ee5\u4e3b\u4f1a'\

'\u6837\u5e74\u60f3\u751f\u540c\u8001\u4e2d\u5341\u4ece'\

'\u81ea\u9762\u524d\u5934\u9053\u5b83\u540e\u7136\u8d70'\

'\u5f88\u50cf\u89c1\u4e24\u7528\u5979\u56fd\u52a8\u8fdb'\

'\u6210\u56de\u4ec0\u8fb9\u4f5c\u5bf9\u5f00\u800c\u5df1'\

'\u4e9b\u73b0\u5c71\u6c11\u5019\u7ecf\u53d1\u5de5\u5411'\

'\u4e8b\u547d\u7ed9\u957f\u6c34\u51e0\u4e49\u4e09\u58f0'\

'\u4e8e\u9ad8\u624b\u77e5\u7406\u773c\u5fd7\u70b9\u5fc3'\

'\u6218\u4e8c\u95ee\u4f46\u8eab\u65b9\u5b9e\u5403\u505a'\

'\u53eb\u5f53\u4f4f\u542c\u9769\u6253\u5462\u771f\u5168'\

'\u624d\u56db\u5df2\u6240\u654c\u4e4b\u6700\u5149\u4ea7'\

'\u60c5\u8def\u5206\u603b\u6761\u767d\u8bdd\u4e1c\u5e2d'\

'\u6b21\u4eb2\u5982\u88ab\u82b1\u53e3\u653e\u513f\u5e38'\

'\u6c14\u4e94\u7b2c\u4f7f\u5199\u519b\u5427\u6587\u8fd0'\

'\u518d\u679c\u600e\u5b9a\u8bb8\u5feb\u660e\u884c\u56e0'\

'\u522b\u98de\u5916\u6811\u7269\u6d3b\u90e8\u95e8\u65e0'\

'\u5f80\u8239\u671b\u65b0\u5e26\u961f\u5148\u529b\u5b8c'\

'\u5374\u7ad9\u4ee3\u5458\u673a\u66f4\u4e5d\u60a8\u6bcf'\

'\u98ce\u7ea7\u8ddf\u7b11\u554a\u5b69\u4e07\u5c11\u76f4'\

'\u610f\u591c\u6bd4\u9636\u8fde\u8f66\u91cd\u4fbf\u6597'\

'\u9a6c\u54ea\u5316\u592a\u6307\u53d8\u793e\u4f3c\u58eb'\

'\u8005\u5e72\u77f3\u6ee1\u65e5\u51b3\u767e\u539f\u62ff'\

'\u7fa4\u7a76\u5404\u516d\u672c\u601d\u89e3\u7acb\u6cb3'\

'\u6751\u516b\u96be\u65e9\u8bba\u5417\u6839\u5171\u8ba9'\

'\u76f8\u7814\u4eca\u5176\u4e66\u5750\u63a5\u5e94\u5173'\

'\u4fe1\u89c9\u6b65\u53cd\u5904\u8bb0\u5c06\u5343\u627e'\

'\u4e89\u9886\u6216\u5e08\u7ed3\u5757\u8dd1\u8c01\u8349'\

'\u8d8a\u5b57\u52a0\u811a\u7d27\u7231\u7b49\u4e60\u9635'\

'\u6015\u6708\u9752\u534a\u706b\u6cd5\u9898\u5efa\u8d76'\

'\u4f4d\u5531\u6d77\u4e03\u5973\u4efb\u4ef6\u611f\u51c6'\

'\u5f20\u56e2\u5c4b\u79bb\u8272\u8138\u7247\u79d1\u5012'\

'\u775b\u5229\u4e16\u521a\u4e14\u7531\u9001\u5207\u661f'\

'\u5bfc\u665a\u8868\u591f\u6574\u8ba4\u54cd\u96ea\u6d41'\

'\u672a\u573a\u8be5\u5e76\u5e95\u6df1\u523b\u5e73\u4f1f'\

'\u5fd9\u63d0\u786e\u8fd1\u4eae\u8f7b\u8bb2\u519c\u53e4'\

'\u9ed1\u544a\u754c\u62c9\u540d\u5440\u571f\u6e05\u9633'\

'\u7167\u529e\u53f2\u6539\u5386\u8f6c\u753b\u9020\u5634'\

'\u6b64\u6cbb\u5317\u5fc5\u670d\u96e8\u7a7f\u5185\u8bc6'\

'\u9a8c\u4f20\u4e1a\u83dc\u722c\u7761\u5174\u5f62\u91cf'\

'\u54b1\u89c2\u82e6\u4f53\u4f17\u901a\u51b2\u5408\u7834'\

'\u53cb\u5ea6\u672f\u996d\u516c\u65c1\u623f\u6781\u5357'\

'\u67aa\u8bfb\u6c99\u5c81\u7ebf\u91ce\u575a\u7a7a\u6536'\

'\u7b97\u81f3\u653f\u57ce\u52b3\u843d\u94b1\u7279\u56f4'\

'\u5f1f\u80dc\u6559\u70ed\u5c55\u5305\u6b4c\u7c7b\u6e10'\

'\u5f3a\u6570\u4e61\u547c\u6027\u97f3\u7b54\u54e5\u9645'\

'\u65e7\u795e\u5ea7\u7ae0\u5e2e\u5566\u53d7\u7cfb\u4ee4'\

'\u8df3\u975e\u4f55\u725b\u53d6\u5165\u5cb8\u6562\u6389'\

'\u5ffd\u79cd\u88c5\u9876\u6025\u6797\u505c\u606f\u53e5'\

'\u533a\u8863\u822c\u62a5\u53f6\u538b\u6162\u53d4\u80cc\u7ec6'

def getEmail():

# 常见域名后缀，可以随意扩展该列表

suffix = ['.com', '.org', '.net', '.cn']

characters = string.ascii\_letters+string.digits+'\_'

username = ''.join((random.choice(characters)

for i in range(random.randint(6,12))))

domain = ''.join((random.choice(characters)

for i in range(random.randint(3,6))))

return username+'@'+domain+random.choice(suffix)

def getTelNo():

return ''.join((str(random.randint(0,9)) for i in range(11)))

def getNameOrAddress(flag):

'''flag=1 表示返回随机姓名，flag=0 表示返回随机地址'''

result = ''

if flag==1:

# 大部分中国人姓名在 2-4 个汉字

rangestart, rangeend = 2, 5

elif flag==0:

# 假设地址在 10-30 个汉字之间

rangestart, rangeend = 10, 31

else:

print('flag must be 1 or 0')

return ''

# 生成并返回随机信息

for i in range(random.randrange(rangestart, rangeend)):

result += random.choice(StringBase)

return result

def getSex():

return random.choice(('男', '女'))

def getAge():

return str(random.randint(18,100))

def main(filename):

with open(filename, 'w', encoding='utf-8') as fp:

# 写入表头

fp.write('Name,Sex,Age,TelNO,Address,Email\n')

# 生成 200 个人的随机信息

for i in range(200):

name = getNameOrAddress(1)

sex = getSex()

age = getAge()

tel = getTelNo()

address = getNameOrAddress(0)

email = getEmail()

line = ','.join([name,sex,age,tel,address,email])+'\n'

fp.write(line)

def output(filename):

with open(filename, 'r', encoding='utf-8') as fp:

for line in fp:

print(line)

if \_\_name\_\_ == '\_\_main\_\_':

filename = 'information.txt'

main(filename)

output(filename)

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

任务16自定义类模拟三维向量及其运算

**一、实验实训目的**

1、了解如何定义一个类。

2、了解如何定义类的私有数据成员和成员方法。

3、了解如何使用自定义类实例化对象。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

**实验内容：**

定义一个三维向量类，并定义相应的特殊方法实现两个该类对象之间的加、减运算（要求支持运算符+、-），实现该类对象与标量的乘、除运算（要求支持运算符\*、/），以及向量长度的计算（要求使用属性实现）。

**实验步骤与参考代码：**

class Vector3:

#构造方法，初始化，定义向量坐标

def \_\_init\_\_(self, x, y, z):

self.\_\_x = x

self.\_\_y = y

self.\_\_z = z

#与两一个向量相加，对应分量相加，返回新向量

def \_\_add\_\_(self, anotherPoint):

x = self.\_\_x + anotherPoint.\_\_x

y = self.\_\_y + anotherPoint.\_\_y

z = self.\_\_z + anotherPoint.\_\_z

return Vector3(x, y, z)

#减去另一个向量，对应分量相减，返回新向量

def \_\_sub\_\_(self, anotherPoint):

x = self.\_\_x - anotherPoint.\_\_x

y = self.\_\_y - anotherPoint.\_\_y

z = self.\_\_z - anotherPoint.\_\_z

return Vector3(x, y, z)

#向量与一个数字相乘，各分量乘以同一个数字，返回新向量

def \_\_mul\_\_(self, n):

x, y, z = self.\_\_x\*n, self.\_\_y\*n, self.\_\_z\*n

return Vector3(x, y, z)

#向量除以一个数字，各分量除以同一个数字，返回新向量

def \_\_truediv\_\_(self, n):

x, y, z = self.\_\_x/n, self.\_\_y/n, self.\_\_z/n

return Vector3(x, y, z)

#查看向量长度，所有分量平方和的平方根

@property

def length(self):

return (self.\_\_x\*\*2 + self.\_\_y\*\*2 + self.\_\_z\*\*2)\*\*0.5

def \_\_str\_\_(self):

return 'Vector3({},{},{})'.format(self.\_\_x,

self.\_\_y,

self.\_\_z)

#用法演示

v1 = Vector3(3, 4, 5)

v2 = Vector3(5, 6, 7)

print(v1+v2)

print(v1-v2)

print(v1\*3)

print(v2/2)

print(v1.length)

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

**任务17 Matplotlib可视化**

**一、实验实训目的**

1.掌握数字特征的计算（A）；

2.掌握相关矩阵计算（A）。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

1. 给定一组单变量数据，分组计算均值、方差、Q1、Q3、偏度、峰度。
2. 给定一组多变量数据，计算相关矩阵。

单样本的数字特征计算

(习题1.4) 从某商店的营业日中随机抽取12天,得日营业额数据为（单位：万元）：

12.5, 17.2, 9.1, 25.4, 31.2, 20, 18.9, 22.8, 21.1, 17.8, 25.1, 27.7

试求样本均值、样本方差、样本变异系数、样本中位数、上样本四分位数、下样本四分位数、样本四分位数间距和极差。

**1. 建数据集**

**Data d4;**

Input **x @@;**

Cards**;**

12.5 17.2 9.1 25.4 31.2 20 18.9 22.8 21.1 17.8 25.1 27.7

**;**

**Run;**

**2. 使用“SAS/ 分析家”菜单**

1. 打开“分析家”界面。

选择SAS界面的级联菜单：“解决方案”⮚“分析”⮚“分析家”。

图形用户界面, 应用程序

描述已自动生成

1. 调出数据文件 Work.D4 。

在界面的空白处，右键弹出菜单，选择级联菜单：“文件”⮚“按SAS名称打开”。依次选择逻辑库和文件对象，分别为“Work”、“D4”，单击“确定”按钮。

图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成图片包含 表格

描述已自动生成

1. 确定分析方法。

选择界面上的级联菜单：“统计”⮚“描述性统计”⮚“分布”

图形用户界面, 应用程序, 网站

描述已自动生成

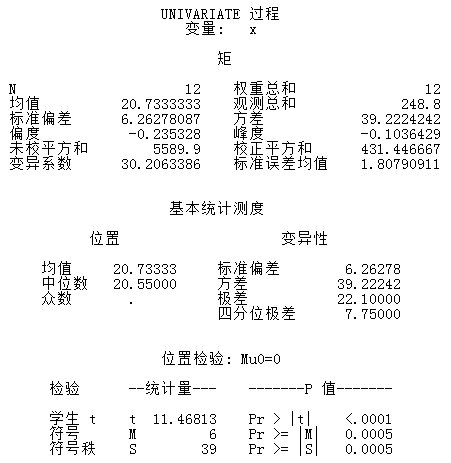
1. 输送变量。

首先选中变量“x”，然后单击“Analyst”按钮。即完成变量的输送，确定变量“x”作为分析变量。单击“OK”。

图形用户界面, 图示

描述已自动生成

1. 得到输出结果



文字图案

中度可信度描述已自动生成

上述使用“分析家”的调用过程等价于程序调用univariate过程：

**proc univariate** data**=d4;**

Var **x;**

**Run;**

**3. 解答**

样本均值 mean =

样本方差 Variance =

样本变异系数 Coeff Variation =

样本上四分位数 75% Q3 =

样本中位数 Median =

样本下四分位数25% Q1 =

极差Range =

样本四分位数间距 Interquartile Range =

相关矩阵计算

(例 3.2.3) 今对31个人进行人体测试，考察或测试的七个指标是：年龄(x1)、体重(x2)、肺活量(x3)、1.5英里跑的时间(x4)、休息时的脉搏(x5)、跑步时的脉搏(x6)和跑步时记录的最大脉搏(x7)。数据列于表3.2.1。现欲对这些指标作一些相关分析。

**1. 对于给定EXCEL数据文件，导入数据建数据集 e323.sas7bdat**

**2. 使用SAS/INSIGHT菜单**

**3. 输出**











**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

**任务18 pandas统计分析基础**

**一、实验实训目的**

1. 掌握单正态总体样本的均值检验（B）；
2. 掌握两独立组样本(服从正态总体)进行均值比较（A）；
3. 掌握两对照组样本(服从正态总体)进行均值比较（A）；
4. 了解两组样本均值差的置信区间计算（C）。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

1. 单样本总体均值的t检验



1. 两独立组样本总体的方差比较



【实验项目设计】

1. 单个pandas统计分析基础；
2. 两独立组样本(服从正态总体)进行均值比较检验；
3. 两对照组样本(服从正态总体)进行均值比较检验。

【实验内容】

一、单个pandas统计分析基础

(习题1.4) 从某商店的营业日中随机抽取12天,得日营业额数据为（单位：万元）：

12.5, 17.2, 9.1, 25.4, 31.2, 20, 18.9, 22.8, 21.1, 17.8, 25.1, 27.7

试检验： μ0 = 10 。

**1. 建数据集**

**Data d4;**

Input **x @@;**

Cards**;**

12.5 17.2 9.1 25.4 31.2 20 18.9 22.8 21.1 17.8 25.1 27.7

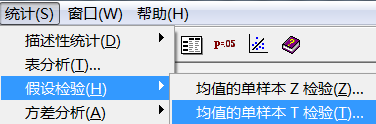
**;**

**Run;**

**2. 使用“SAS/ 分析家”菜单**

1. 打开“分析家”界面。
2. 调出数据文件 Work.D4 。
3. 确定分析方法。

选择界面上的级联菜单：“统计”⮚“假设检验”⮚“均值的单样本T检验”



1. 输送变量。

确定变量“x”为分析变量。

1. 得到输出结果

表格

描述已自动生成

**3. 解答**

H0: μ0=10 H1: μ0≠10

检验方法：t检验

统计量取值: t=\_\_\_\_

检验p值=\_\_\_\_\_\_\_

结论：拒绝/接受H0.

1. 两独立组样本(服从正态总体)进行均值比较

(习题1.22) 为比较甲、乙两种安眠药的疗效，将20名患者分成两组，每组10人，假定服药后延长的睡眠时间都近似服从正态分布且方差相等，测得其数据如下：

1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 5.5, 1.6, 4.6, 3.4

0.7, -1.6, -0.2, -1.2, -0.1, 3.4, 3.7, 0.8, 0, 2

试问在显著性水平a=0.05下检验两种安眠药的疗效有无显著性差异，p值是多少？

**1. 建数据集**

前面的“分析家”界面不要关闭，切换到程序编辑窗口，输入程序。

Data d22;

Input x y;

Cards;

1.9 0.7

0.8 -1.6

1.1 -0.2

0.1 -1.2

-0.1 -0.1

4.4 3.4

5.5 3.7

1.6 0.8

4.6 0

3.4 2

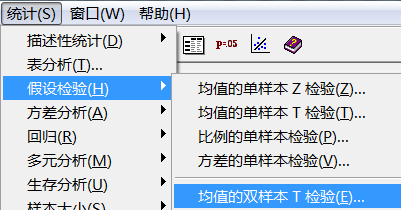
;

Run;

**2. 使用“SAS/分析家”菜单**

1. 回到“分析家”界面，调出数据文件 work.d22
2. 确定分析方法。

选择级联菜单：“统计”⮚“假设检验”⮚“均值的双样本T检验”



1. 设置数据集work.d22中，各样本分组数据对应的变量个数：“Two variables” 。

文本

描述已自动生成

1. 输送变量。

将变量“x”输送给 “group1”，“y”输送给 “group2”，单击“OK”

图示

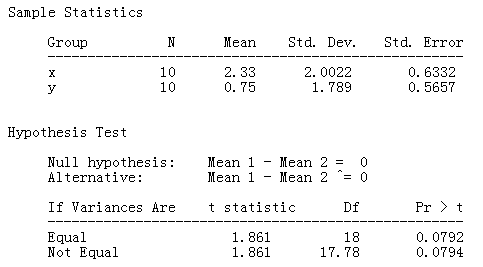
描述已自动生成图形用户界面

描述已自动生成

图示

描述已自动生成

1. 输出结果如下



1. 涉及到两独立组的方差，继续检验两组方差是否相等。

回到“分析家”界面上，选择级联菜单“统计”⮚“假设检验”⮚“方差的双样本T检验”

图形用户界面, 文本, 应用程序

描述已自动生成

1. 输送变量。

将变量“x”输送给 “group1”，“y”输送给 “group2”，单击“OK”。

1. 输出结果如下

一些文字和图片的手机截图

中度可信度描述已自动生成

**3. 解答**

H0: σ1=σ2 H1: σ1≠σ2

检验方法：F检验

统计量取值:F=\_\_\_\_

检验p值=\_\_\_\_\_\_\_

结论：拒绝/接受H0.两独立组的方差是/否相等。

H0: μ1=μ2 H1: μ1≠μ2

检验方法：t检验

统计量取值: t=\_\_\_\_

检验p值=\_\_\_\_\_\_\_

结论：拒绝/接受H0. 两种安眠药的疗效有/无显著性差异

三、两对照组样本(服从正态总体)进行均值比较

（习题1.24） 甲、乙两种稻种分别播种在10块试验田中，每块试验田甲、乙稻种各种一半。假设同一块田中两种作物的产量之差服从正态分布。现获10块田的成对产量如下表所示（单位：公斤），试问两种稻种产量有无显著性差异（a=0.05），p值是多少？

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 甲种 | 140 | 137 | 136 | 140 | 145 | 148 | 140 | 135 | 144 | 141 |
| 乙种 | 135 | 138 | 125 | 140 | 128 | 131 | 130 | 140 | 131 | 125 |

**1. 建数据集**

Data d24;

Input x y @@ ;

Cards;

140 135 137 138 136 125 140 140 145 128

148 131 140 130 135 140 144 131 141 125

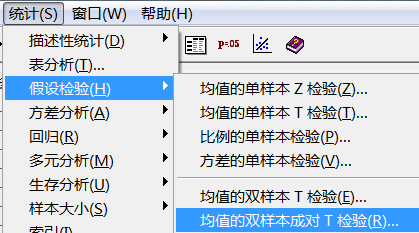
;

Run;

**2. 使用“SAS/分析家”菜单**

1. 回到“分析家”界面，调出数据文件 work.d24
2. 确定分析方法。

选择级联菜单：“统计”⮚“假设检验”⮚“均值的双样本成对T检验”



1. 输送变量。

设置各分组数据对应的变量个数：“Two variables”，并输送变量“x”“y”，单击“OK”，得到输出结果如下：

手机屏幕截图

描述已自动生成

**3. 解答**

H0: μ1=μ2 H1: μ1≠μ2

检验方法：t检验

统计量取值:t=\_\_\_\_

检验p值=\_\_\_\_\_\_\_

结论：拒绝/接受H0，两种稻种产量有/无显著性差异.

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

**任务19 pandas数据预处理**

**一、实验实训目的**

* 1. 掌握卡方检验（A）；
  2. 熟悉符合检验、符号秩和检验、秩和检验。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

1. 对列联表，检验行变量与列变量的关联性；
2. 给定一组未知总体的样本数据，检验μ=μ0；
3. 给定两组未知总体的样本数据，检验μ1=μ2。

对列联表，检验行变量与列变量的关联性

(习题4.7) 下表为某药治疗感冒效果的3x3列表。

疗效\年龄 儿童 成年 老年 合计

显著 58 38 32 128

一般 28 44 45 117

较差 23 18 14 55

合计 109 100 91 300

试问：疗效与年龄是否关联（a=0.05）？

**1. 建数据集**

**Data** d0407;

Do effect='显著', '一般','较差';

Do age='儿童','成年','老年';

input num @@;

output;

end;

end;

Cards;

58 38 32

28 44 45

23 18 14

;

**Run**;

**2. 使用“SAS/ 分析家”菜单**

1. 打开“分析家”界面。
2. 调出数据文件 Work.D0407。
3. 确定分析方法。

选择界面上的级联菜单：“统计”⮚“表分析”

图形用户界面, 文本

描述已自动生成

得到列联表的空白界面。

图形用户界面

描述已自动生成

1. 输送变量。

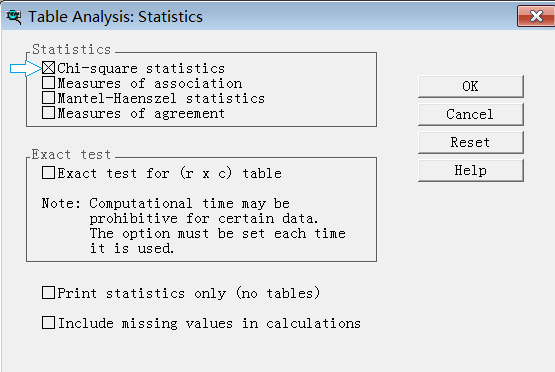
选中变量“effect”，单击“Row”按钮；选中变量“age”，单击“Column”按钮；选中变量“num”，单击“Cell Counts”按钮。完成变量输送。显示如下。

图形用户界面

描述已自动生成

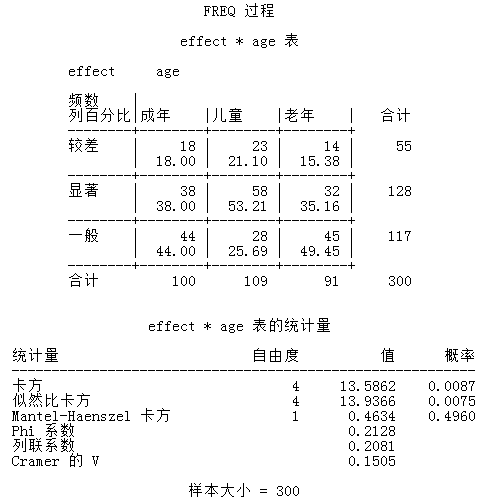
1. 选择附加的统计方法。

在列联表分析（Table Analysis）界面的右边下角，点击“statistics”按钮，打开选项设置页。勾选“chi-square statistics”。如图所示。单击“OK”按钮。



值得一提的是，卡方检验默认使用近似计算，目的是为了节省时间。当然也可以勾选中间的“Exact test”选项卡，来指定精确检验，只是某些时候的运行时间无法控制。

1. 得到输出结果



**3. 解答**

H0：疗效与年龄独立。

检验统计量及其分布：

统计量=

检验p值=

结论：

二、对未知总体的一组样本数据，进行均值的pandas数据预处理 （H0：μ=μ0）

(习题 4.9 ) 从某企业中随机抽取12名职工，调查他们的年抽入，数据如下（单位：万元）：

5.42 5.65 5.64 5.13 4.69 5.13 4.81 4.37 5.82 6.31 4.78 5.52

试用符号检验对该企业职工年收入的中位数是否显著超过了5万元进行符号检验（a=0.05）。

1. **建数据集**

前面的“分析家”界面不要关闭，切换到程序窗口，运行以下程序。

**Data** d0409;

input income @@;

d = income - **5**;

cards;

5.42 5.65 5.64 5.13 4.69 5.13

4.81 4.37 5.82 6.31 4.78 5.52

;

**run**;

**2. 使用“SAS/ 分析家”菜单**

1. 在未关闭的“分析家”界面中，调出数据文件 Work.D0409。
2. 打开对应于univariate程序功能的菜单分析界面。

选择界面上的级联菜单：“统计”⮚“描述性统计”⮚“分布”

1. 输送差值变量“d”。选中差值变量“d”，单击Analysis框。单击“OK”。
2. 得到输出结果。

图示, 示意图

描述已自动生成

**3. 解答**

H0：m=5 H1：m>5

①符号检验

检验统计量及其分布：

统计量 s+= 2+**E(S+)**=8

检验p值= 0.3877 **/ 2**=0.194 （单边检验，故概率除以2）

结论：接受H0。

②符号秩检验（前提：假定总体的分布式对称的）

检验统计量代表差值变量d中取值为正的数据项秩总和。

统计量 T+= 19 + **E(T+)** = 19 + 12(12+1)/4 = 19 + 39 =58,

检验p值= 0.1455/2=0.073（单边检验，故概率除以2）

结论：接受H0。

三、两组未知总体的样本数据，pandas数据预处理总体均值

（习题4.12） 在甲、乙两台同型梳棉机上，进行纤维转移率试验，除机台外其他工艺条件都相同，经试验得两个容量不同的纤维转移率样本数据如下表：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 甲 | 8.635 | 9.071 | 9.231 | 8.797 | 9.87 | 10.325 |  |  |  |
| 乙 | 7.342 | 7.436 | 8.03 | 8.946 | 9.131 | 6.823 | 8.965 | 8.696 | 8.345 |

试用秩和检验，对纤维转移率而言，这两台机器是否存在显著的机台差异（a=0.05）.

1. **建数据集**

前面的“分析家”界面不要关闭，切换到程序窗口，运行以下程序。

**data** d04012;

input x class $;

cards;

8.635 甲

9.071 甲

9.231 甲

8.797 甲

9.87 甲

10.325 甲

7.342 乙

7.436 乙

8.03 乙

8.946 乙

9.131 乙

6.823 乙

8.965 乙

8.696 乙

8.345 乙

;

**run**;

1. **运行npar1way 过程**

**proc** **npar1way** data=d04012 wilcoxon;

class c;

var x;

**run**;

**3. 得到输出结果**

文本

描述已自动生成

**4. 解答**

H0：两台机器没有显著的差异（即μ1=μ2） H1：两台机器存在显著的差异(即μ1≠μ2)

T=67，(统计量 T代表甲台机器的样本数据在混合数据中的秩和)，

检验p值=0.0292 （双边检验，故选用z检验的 “Two – sided Pr > |Z| ”）,

拒绝H0，

认为两台机器存在显著差异。

**【思考题】**

问题1说出pandas数据预处理和参数检验的区别。

问题2说出pandas数据预处理中，都有哪几类问题有需要以引入差值变量d。

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

无

**任务20 航空客户价值分析**

**一、实验实训目的**

掌握多元线性回归方程的拟合、检验（A）。

**二、实验实训要求**

操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**

1. 给定样本数据，进行一元线性回归拟合、检验；
2. 给定样本数据，进行多元线性回归拟合、检验。
3. 带虚拟变量（哑元变量）的线性回归。

【实验内容】

一、一元航空客户价值分析

（习题5.2） 关于汽车轮胎的等级和承载能力的一个样本统计资料如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 承载能力 y | 853 | 1047 | 1135 | 1201 | 1235 | 1356 | 1389 | 1433 | 2039 |
| 轮胎等级 x | 75 | 82 | 85 | 87 | 88 | 91 | 92 | 93 | 105 |

(1) 用汽车轮胎的等级作自变量，画出这些数据的散点图；

(2) 利用最小二乘法，求出估计回归函数；

**1. 建数据集**

**Data d0604;**

Input **y x1 @@;**

Cards**;**

**补充**

**;**

**Run;**

**2. 使用“SAS/ 分析家”菜单**

1. 打开“分析家”界面。调出数据文件 Work.D0602 。
2. 绘制散点图。

单击工具栏“散点图”图标

图形用户界面

描述已自动生成

弹出散点图页面：

图形用户界面

描述已自动生成

（2.1）输送变量。

（2.2）设置星号为散点显示符号。

单击“display”按钮，弹出页面如下，

图形用户界面, 文本, 应用程序

描述已自动生成

单击黑色下箭头，弹出列表中单击“star”。单击“OK”返回。

1. 拟合回归函数。步骤同实验内容二。略。

二、多元航空客户价值分析

（习题6.4）关下表是30家计算机硬件设备公司的每股价格( y )、每股账面价值( x1 )和每股资本收益率( x2 )的统计资料：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| y | x1 | x2 | y | x1 | x2 |
| 12.31 | 4.94 | -49.7 | 28.75 | 6.35 | 15.1 |
| 21.75 | 9.46 | -71.8 | 30.5 | 3.78 | 11.8 |
| 11 | 4.95 | 17.2 | 27.19 | 12.59 | 9.8 |
| 3.25 | 4.33 | 5.1 | 16.31 | 3.64 | 28.3 |
| 65.5 | 9.58 | 20.8 | 11.88 | 3.56 | 4 |
| 35.94 | 8.46 | 13.3 | 33 | 10.03 | 16.5 |
| 82.06 | 2.33 | 74.5 | 28.19 | 10.64 | 3.3 |
| 15 | 7.35 | -11.9 | 27.44 | 9.12 | -4.3 |
| 43 | 22.4 | -12.9 | 15.13 | 6.15 | 16.1 |
| 14.25 | 4.11 | 18.8 | 55.5 | 22.38 | 11.1 |
| 16.25 | 6.83 | 10.7 | 48 | 6.4 | 26.2 |
| 12.88 | 9.13 | 9 | 34.24 | 9.49 | 8.7 |
| 39.13 | 6.07 | 28.8 | 38.94 | 10.25 | 14.3 |
| 61.5 | 14.14 | 18.7 | 11.31 | 0.68 | 1.6 |
| 101.38 | 20.12 | 29.9 | 14.63 | 3.48 | 24.3 |

1. 试求估计回归函数；
2. 检验H0:0=1=0（a=0.05）；
3. 试求复判定系数R2；
4. 试求0和1的95%置信区间；
5. 对于一家每股账面价值x1= 9.48和每股资本收益率x2= 17.5的计算机，硬件设备公司，试对它的平均每股价格作出点估计并对其每股价格作出点预测；
6. 在 (5) 中，试求平均每股价格的95%置信区间和每股价格的0.95置信区间.

**1. 建数据集**

**Data d0604;**

Input **y x1 x2 @@;**

Cards**;**

12.31 4.94 -49.7 28.75 6.35 15.1

21.75 9.46 -71.8 30.5 3.78 11.8

11 4.95 17.2 27.19 12.59 9.8

3.25 4.33 5.1 16.31 3.64 28.3

65.5 9.58 20.8 11.88 3.56 4

35.94 8.46 13.3 33 10.03 16.5

82.06 2.33 74.5 28.19 10.64 3.3

15 7.35 -11.9 27.44 9.12 -4.3

43 22.4 -12.9 15.13 6.15 16.1

14.25 4.11 18.8 55.5 22.38 11.1

16.25 6.83 10.7 48 6.4 26.2

12.88 9.13 9 34.24 9.49 8.7

39.13 6.07 28.8 38.94 10.25 14.3

61.5 14.14 18.7 11.31 0.68 1.6

101.38 20.12 29.9 14.63 3.48 24.3

**;**

**Run;**

**2. 使用“SAS/ 分析家”菜单**

1. 打开“分析家”界面。调出数据文件 Work.D0604 。
2. 打开（多元）线性回归界面。

单击工具栏“(多元)线性回归”图标：

图形用户界面, 应用程序, Teams

描述已自动生成

1. 设置（多元）线性回归界面，如图。

图形用户界面, 应用程序, Word

描述已自动生成

（3.1）输送变量。（此步骤操作必选，余下步骤可选）

输送y为Dependent变量；同时选中x1、x2设置为Explanatory变量。

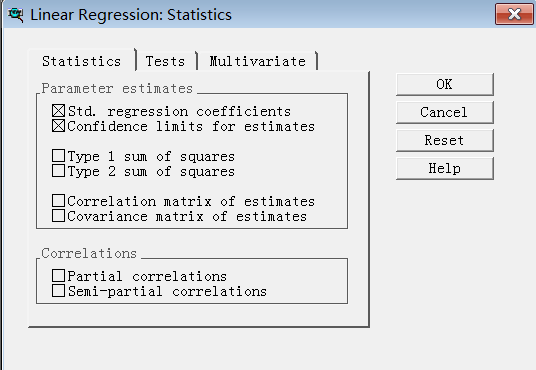
（3.2）求回归系数求0和1的95%置信区间。

单击Statistics按钮，弹出页面：

文本

描述已自动生成

在Parameter estimates选项框中，勾选前两项。如图。即设置输出标准回归系数和参数估计置信区间。



（3.3）新采样点的点预测。

依据题目为例：x1= 9.48，x2= 17.5。

对平均每股价格作出点估计并对其每股价格作出点预测。

* 切换到程序编辑窗口，建立待预测数据集temp并运行程序。

Data temp;

x1= 9.48;

x2= 17.5;

y=.;

run;

* 切换到多元线性回归界面，单击Predictions按钮，弹出页面：

图形用户界面, 应用程序

描述已自动生成

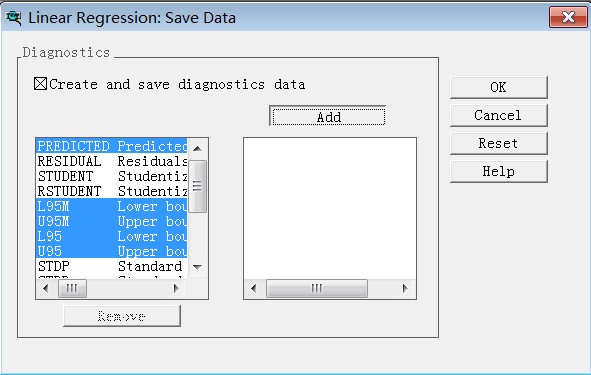
在”预测输入”选项框中，勾选“Predict additional data”，单击实黑色右箭头，将弹出“选择成员”页面，选择新建的待预测数据集work.temp。然后，在“预测输出”选项框中，勾选“List predictions”，将显示预测结果列表。单击“OK”返回。

* 多元线性回归界面，单击Save data按钮，弹出页面：

图形用户界面, 应用程序

描述已自动生成

勾选“Create and save diagnostics data”选项。



按住键盘shift按钮，依次单击Predicted、L95M、U95M、L95、U95（含义分别是：预测值，平均y值的95%置信下限、上限，预测y值的95%置信下限、上限），单击界面“Add”按钮，完成预测统计量的添加。单击OK返回。

（3.4）完成多元线性回归界面设置，单击OK按钮。得到输出结果。

手机屏幕截图

描述已自动生成

1. 保存数据的查看。

在SAS资源管理器中，双击新出现的文件夹 \_proj\_ ,

图形用户界面, 应用程序

描述已自动生成 图形用户界面, 应用程序

描述已自动生成

双击Score，即可查看预测输出保存数据。

**3. 解答**

略。

1. 带虚拟变量的多元航空客户价值分析

(习题6.4) 一位经济学家想研究采取某项改革措施所需时间（y）对保险公司规模（x1，单位：百万美元）和保险公司类型（互助公司和股份公司）的关系。因变量的计量是第一个公司采纳这项革新和给定公司采纳这项革新在时间上先后间隔的月数。令虚拟变量X2=1 互助公司；X2=0 股份公司。

考虑回归模型：

Yi=0+1xi1+2xi2+i

10个互助公司和10个股份公司的有关调查数据见下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 公司 | y | x1 | 公司类型 | 公司 | Y | x1 | 公司类型 |
| 1 | 17 | 151 | 互助 | 11 | 28 | 164 | 股份 |
| 2 | 26 | 92 | 互助 | 12 | 15 | 272 | 股份 |
| 3 | 21 | 175 | 互助 | 13 | 11 | 295 | 股份 |
| 4 | 30 | 31 | 互助 | 14 | 38 | 68 | 股份 |
| 5 | 22 | 104 | 互助 | 15 | 31 | 85 | 股份 |
| 6 | 0 | 277 | 互助 | 16 | 21 | 224 | 股份 |
| 7 | 12 | 210 | 互助 | 17 | 20 | 166 | 股份 |
| 8 | 19 | 120 | 互助 | 18 | 13 | 305 | 股份 |
| 9 | 4 | 290 | 互助 | 19 | 30 | 124 | 股份 |
| 10 | 16 | 238 | 互助 | 20 | 14 | 246 | 股份 |

1. 试求估计回归函数，并写出对每一公司类型的估计回归函数；
2. 解释回归系数的含义；
3. 检验H0:1=2=0（a=0.05）；
4. 检验验H0:1 =0 和H0: 2=0（a=0.05）；
5. 试求1和2的95%置信区间。

**实验步骤同上。略。.**

**.四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**

**任务21 预测分析**

**一、实验实训目的**

掌握预测分析不同方法

待判样品的距离判别法和贝叶斯判别法（A）。

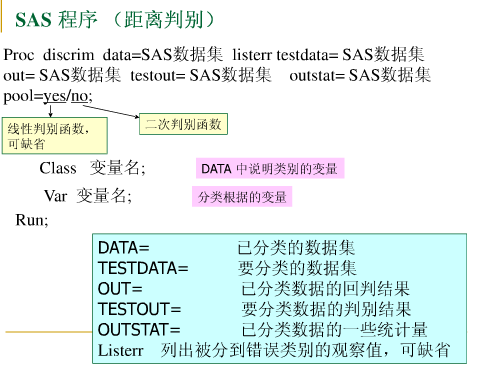
**二、实验实训要求**

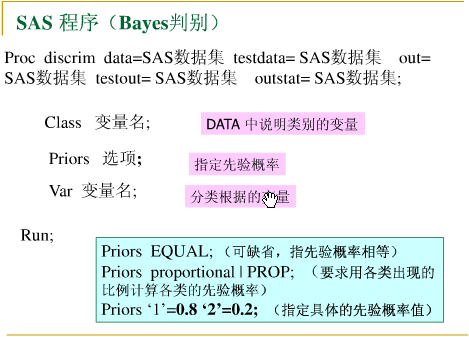
操作系统：Windows 7版本及以上

Python环境：Python 3.7

开发工具：1.IDLE 2.Anaconda3 3.sublime text 3

**三、实验实训步骤**





【实验项目设计】

1. 距离判别;
2. 贝叶斯判别。

【实验内容】

（习题8.4） 根据经验，今天和昨天的湿度差*x* 1及今天的压温差（气压与温度之差）*x* 2是预报明天是否下雨的两个重要因素。现收集到一批样本数据列于下表。

|  |  |  |  |
| --- | --- | --- | --- |
| 1（雨天） | | 2（非雨天） | |
| x1（湿度差） | x2（压温差） | x1（湿度差） | x2（压温差） |
| -1.9 | 3.2 | 0.2 | 6.2 |
| -6.9 | 10.4 | -0.1 | 7.5 |
| 5.2 | 2 | 0.4 | 14.6 |
| 5 | 2.5 | 2.7 | 8.3 |
| 7.3 | 0 | 2.1 | 0.8 |
| 6.8 | 12.7 | -4.6 | 4.3 |
| 0.9 | -15.4 | -1.7 | 10.9 |
| -12.5 | -2.5 | -2.6 | 13.1 |
| 1.5 | 1.3 | 2.6 | 12.8 |
| 3.8 | 6.8 | -2.8 | 10 |
|  |  |  |  |

今测得*x* 1=0.6， *x* 2=3.0，假定两组的协方差阵相等。

1. 试用距离判别法预报明天是否下雨，用（8.2.7）式即回判法来估计误判概率；

解答：明天下雨/不下雨\_明天下雨\_。

误判率的回判法估计： 0.2 ,  0.1 。

1. 假定两组的均服从二元正态分布，且根据其他信息及经验给出先验概率*p*1= 0.3，*p*2= 0.7，试用贝叶斯判别法预报明天是否下雨；

解答：明天下雨/不下雨\_\_\_\_\_不下雨\_\_\_\_\_\_。

1. 假如你现考虑是否为明天安排一项活动，该活动在时间上有紧迫性，但又不太适合在雨天进行，并认为C(2|1)= 3C(1|2)，那么你今天是否应该安排项这活动呢？

判别规则为：



,

解答：安排/不安排\_\_不安排\_\_\_\_\_\_\_\_\_。

8.4 程序附录：

**Data** d0804;

Input group x1 x2 @@;

Cards;

1 -1.9 3.2 2 0.2 6.2 1 -6.9 10.4 2 -0.1 7.5

1 5.2 2 2 0.4 14.6 1 5 2.5 2 2.7 8.3

1 7.3 0 2 2.1 0.8 1 6.8 12.7 2 -4.6 4.3

1 0.9 -15.4 2 -1.7 10.9 1 -12.5 -2.5 2 -2.6 13.1

1 1.5 1.3 2 2.6 12.8 1 3.8 6.8 2 -2.8 10

;

Run;

**Data** t\_d;

x1=**0.6**; x2=**3.0**; group=**.**;

run;

**Proc** **discrim** data=d0804 testdata=t\_d pool=yes list testlist listerr ;

Class group;

Var x1 x2;

**run**;

**Proc** **discrim** data=d0804 testdata=t\_d pool=yes testlist ;

Class group;

Var x1 x2;

Priors '1'=**0.3** '2'=**0.7**;

**run**;

习题8.5) 对28名一级和25名健将级标枪运动员测试了6个影响标枪成绩的训练项

目，这些训练项目（成绩）为：30 米跑（*x* 1） ，投掷小球（*x* 2） ，挺举重量（*x* 3 ） ，抛实心球（*x* 4 ） ，前抛铅球（*x* 5 ），五级跳（*x* 6 ），全部数据列于下表。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | X1 | X2 | X3 | X4 | X5 | X6 | 组别 |
| 1 | 3.6 | 4.3 | 82.3 | 70 | 90 | 18.52 | 1 |
| 2 | 3.3 | 4.1 | 87.48 | 80 | 100 | 18.48 | 1 |
| 3 | 3.3 | 4.22 | 87.74 | 85 | 115 | 18.56 | 1 |
| 4 | 3.21 | 4.05 | 88.6 | 75 | 100 | 19.1 | 1 |
| 5 | 3.1 | 4.38 | 89.98 | 95 | 120 | 20.14 | 1 |
| 6 | 3.2 | 4.9 | 89.1 | 85 | 105 | 19.44 | 1 |
| 7 | 3.3 | 4.2 | 89 | 75 | 85 | 19.17 | 1 |
| 8 | 3.5 | 4.5 | 84.2 | 80 | 100 | 18.8 | 1 |
| 9 | 3.7 | 4.6 | 82.1 | 70 | 85 | 17.68 | 1 |
| 10 | 3.4 | 4.4 | 90.18 | 75 | 100 | 19.14 | 1 |
| 11 | 3.6 | 4.3 | 82.1 | 70 | 90 | 18.1 | 1 |
| 12 | 3.6 | 4.5 | 82 | 55 | 70 | 17.4 | 1 |
| 13 | 3.6 | 4.2 | 82.2 | 70 | 90 | 18.12 | 1 |
| 14 | 3.4 | 4.2 | 85.4 | 85 | 100 | 18.66 | 1 |
| 15 | 3.3 | 4.3 | 90.1 | 80 | 100 | 19.86 | 1 |
| 16 | 3.12 | 4.2 | 89 | 85 | 100 | 20 | 1 |
| 17 | 3.1 | 4.2 | 90.2 | 85 | 115 | 20.8 | 1 |
| 18 | 3.6 | 4.2 | 81.96 | 65 | 80 | 17.2 | 1 |
| 19 | 3.7 | 4.4 | 81 | 80 | 95 | 17 | 1 |
| 20 | 3.3 | 4.3 | 90 | 80 | 110 | 19.8 | 1 |
| 21 | 3.8 | 4.09 | 80 | 60 | 80 | 16.89 | 1 |
| 22 | 3.7 | 4.3 | 83.9 | 85 | 100 | 18.76 | 1 |
| 23 | 3.5 | 4.2 | 85.4 | 85 | 100 | 18.7 | 1 |
| 24 | 3.4 | 4.1 | 86.7 | 85 | 110 | 18.5 | 1 |
| 25 | 3.3 | 4.1 | 88.1 | 75 | 85 | 18.96 | 1 |
| 26 | 3.7 | 4.1 | 84.1 | 70 | 95 | 18.7 | 1 |
| 27 | 3.6 | 4.3 | 82 | 70 | 90 | 18.4 | 1 |
| 28 | 3.2 | 4.2 | 89.2 | 85 | 115 | 19.88 | 1 |
| 29 | 3.4 | 4 | 103 | 95 | 110 | 24.8 | 2 |
| 30 | 3.3 | 4.5 | 118 | 90 | 120 | 25.7 | 2 |
| 31 | 3.1 | 4.5 | 105 | 85 | 110 | 25.1 | 2 |
| 32 | 3.8 | 4.1 | 104.53 | 80 | 100 | 24.98 | 2 |
| 33 | 3 | 4.2 | 112 | 95 | 125 | 25.35 | 2 |
| 34 | 3.9 | 3.7 | 98.2 | 85 | 90 | 21.8 | 2 |
| 35 | 3.5 | 4.1 | 98.7 | 90 | 120 | 22.78 | 2 |
| 36 | 3.1 | 3.9 | 98.2 | 60 | 90 | 21.98 | 2 |
| 37 | 3.3 | 3.9 | 109 | 100 | 120 | 25.3 | 2 |
| 38 | 3.1 | 3.95 | 98.4 | 95 | 115 | 25.2 | 2 |
| 39 | 3.14 | 3.9 | 95.3 | 90 | 110 | 21.42 | 2 |
| 40 | 3.6 | 4.3 | 93.6 | 75 | 85 | 20.84 | 2 |
| 41 | 3.12 | 3.9 | 95.8 | 80 | 105 | 21.8 | 2 |
| 42 | 3 | 3.9 | 93.8 | 85 | 90 | 21.08 | 2 |
| 43 | 3.4 | 3.91 | 96.3 | 110 | 120 | 21.98 | 2 |
| 44 | 3.63 | 3.78 | 98.56 | 85 | 120 | 22.36 | 2 |
| 45 | 3.3 | 3.98 | 97.4 | 85 | 100 | 22.34 | 2 |
| 46 | 3.3 | 4.4 | 112 | 75 | 110 | 25.1 | 2 |
| 47 | 3.5 | 4.1 | 107.7 | 87.5 | 110 | 25.1 | 2 |
| 48 | 3.4 | 4.2 | 92.1 | 80 | 120 | 22.16 | 2 |
| 49 | 3.6 | 4.1 | 99.48 | 85 | 120 | 23.1 | 2 |
| 50 | 3.1 | 4.4 | 116 | 75 | 110 | 25.3 | 2 |
| 51 | 3.12 | 4 | 102.7 | 80 | 110 | 24.68 | 2 |
| 52 | 3.6 | 4.1 | 115 | 85 | 115 | 23.7 | 2 |
| 53 | 3.5 | 4.3 | 97.8 | 75 | 100 | 24.1 | 2 |

另有14名未定级的运动员也测试了同样6个项目，数据列表如下。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | X1 | X2 | X3 | X4 | X5 | X6 |
| 1 | 3.5 | 4.1 | 85.3 | 75 | 105 | 18.65 |
| 2 | 3.4 | 4.4 | 85.4 | 75 | 95 | 18.6 |
| 3 | 3.6 | 4.3 | 85.36 | 75 | 90 | 18.6 |
| 4 | 3.6 | 4.1 | 83.7 | 75 | 105 | 18.6 |
| 5 | 3.2 | 4.1 | 89.35 | 75 | 95 | 20.28 |
| 6 | 3.4 | 4.15 | 86.28 | 60 | 77.5 | 18.9 |
| 7 | 3.6 | 4.2 | 84.1 | 80 | 100 | 18.7 |
| 8 | 3.1 | 4.1 | 98 | 95 | 130 | 22.3 |
| 9 | 3 | 4.1 | 122 | 100 | 115 | 27.1 |
| 10 | 3.2 | 4.3 | 92.68 | 80 | 105 | 20.68 |
| 11 | 3.1 | 4.2 | 91.76 | 85 | 100 | 22.2 |
| 12 | 3.3 | 4.2 | 98.4 | 65 | 100 | 22.86 |
| 13 | 3.3 | 4.6 | 92 | 80 | 195 | 23.07 |
| 14 | 3.4 | 4.3 | 97.36 | 75 | 110 | 22.12 |

假定两组数据均来自于电影正态总体，且C(2|1)= C(1|2)。

1. 对14名未定级的运动员，假设*p*1=*p*2，试在和的两组情形下分别都他们归属何组作出贝叶斯判别；

① 

判入一级的运动员编号：{1,3,5,6,7,9,11,12,13}

判入二级的运动员编号：{ 2,4,8,10,14　}

②

判入一级的运动员编号：{1,3,5,6,7,9,11,13 }

判入二级的运动员编号：{2,4,8,10,12,14 }

1. 试按（8.2.7）和（8.2.8）两式分别对(1)的误判概率作出估计；

① 

回判法的误判率： 0 ,  0 。

刀切法的误判率： 0 ,  0 。

②

回判法的误判率： 0 ,  0 .08 。

刀切法的误判率： 0 ,  0 。

1. 假设，*p*1=0.8，*p*2=0.2，试对14名未定级的运动员的归属作出贝叶斯判别。

判入一级的运动员编号：{1,2,3,5,6,7,9,11,12,13}

判入一级的运动员编号：{4,8,10,14}

8.5程序附录：

Data d0805;

Input no x1-x6 group @@;

Cards;

**1** **3.6** **4.3** **82.3** **70** **90** **18.52** **1** **29** **3.4** **4** **103** **95** **110** **24.8** **2**

**2** **3.3** **4.1** **87.48** **80** **100** **18.48** **1** **30** **3.3** **4.5** **118** **90** **120** **25.7** **2**

**3** **3.3** **4.22** **87.74** **85** **115** **18.56** **1** **31** **3.1** **4.5** **105** **85** **110** **25.1** **2**

**4** **3.21** **4.05** **88.6** **75** **100** **19.1** **1** **32** **3.8** **4.1** **104.53** **80** **100** **24.98** **2**

**5** **3.1** **4.38** **89.98** **95** **120** **20.14** **1** **33** **3** **4.2** **112** **95** **125** **25.35** **2**

**6** **3.2** **4.9** **89.1** **85** **105** **19.44** **1** **34** **3.9** **3.7** **98.2** **85** **90** **21.8** **2**

**7** **3.3** **4.2** **89** **75** **85** **19.17** **1** **35** **3.5** **4.1** **98.7** **90** **120** **22.78** **2**

**8** **3.5** **4.5** **84.2** **80** **100** **18.8** **1** **36** **3.1** **3.9** **98.2** **60** **90** **21.98** **2**

**9** **3.7** **4.6** **82.1** **70** **85** **17.68** **1** **37** **3.3** **3.9** **109** **100** **120** **25.3** **2**

**10** **3.4** **4.4** **90.18** **75** **100** **19.14** **1** **38** **3.1** **3.95** **98.4** **95** **115** **25.2** **2**

**11** **3.6** **4.3** **82.1** **70** **90** **18.1** **1** **39** **3.14** **3.9** **95.3** **90** **110** **21.42** **2**

**12** **3.6** **4.5** **82** **55** **70** **17.4** **1** **40** **3.6** **4.3** **93.6** **75** **85** **20.84** **2**

**13** **3.6** **4.2** **82.2** **70** **90** **18.12** **1** **41** **3.12** **3.9** **95.8** **80** **105** **21.8** **2**

**14** **3.4** **4.2** **85.4** **85** **100** **18.66** **1** **42** **3** **3.9** **93.8** **85** **90** **21.08** **2**

**15** **3.3** **4.3** **90.1** **80** **100** **19.86** **1** **43** **3.4** **3.91** **96.3** **110** **120** **21.98** **2**

**16** **3.12** **4.2** **89** **85** **100** **20** **1** **44** **3.63** **3.78** **98.56** **85** **120** **22.36** **2**

**17** **3.1** **4.2** **90.2** **85** **115** **20.8** **1** **45** **3.3** **3.98** **97.4** **85** **100** **22.34** **2**

**18** **3.6** **4.2** **81.96** **65** **80** **17.2** **1** **46** **3.3** **4.4** **112** **75** **110** **25.1** **2**

**19** **3.7** **4.4** **81** **80** **95** **17** **1** **47** **3.5** **4.1** **107.7** **87.5** **110** **25.1** **2**

**20** **3.3** **4.3** **90** **80** **110** **19.8** **1** **48** **3.4** **4.2** **92.1** **80** **120** **22.16** **2**

**21** **3.8** **4.09** **80** **60** **80** **16.89** **1** **49** **3.6** **4.1** **99.48** **85** **120** **23.1** **2**

**22** **3.7** **4.3** **83.9** **85** **100** **18.76** **1** **50** **3.1** **4.4** **116** **75** **110** **25.3** **2**

**23** **3.5** **4.2** **85.4** **85** **100** **18.7** **1** **51** **3.12** **4** **102.7** **80** **110** **24.68** **2**

**24** **3.4** **4.1** **86.7** **85** **110** **18.5** **1** **52** **3.6** **4.1** **115** **85** **115** **23.7** **2**

**25** **3.3** **4.1** **88.1** **75** **85** **18.96** **1** **53** **3.5** **4.3** **97.8** **75** **100** **24.1** **2**

**26** **3.7** **4.1** **84.1** **70** **95** **18.7** **1**

**27** **3.6** **4.3** **82** **70** **90** **18.4** **1**

**28** **3.2** **4.2** **89.2** **85** **115** **19.88** **1**

;

Run;

Data t\_d;

Input no x1-x6 @@;

group=. ;

Cards;

**1** **3.5** **4.1** **85.3** **75** **105** **18.65** **8** **3.1** **4.1** **98** **95** **130** **22.3**

**2** **3.4** **4.4** **85.4** **75** **95** **18.6** **9** **3** **4.1** **122** **100** **115** **27.1**

**3** **3.6** **4.3** **85.36** **75** **90** **18.6** **10** **3.2** **4.3** **92.68** **80** **105** **20.68**

**4** **3.6** **4.1** **83.7** **75** **105** **18.6** **11** **3.1** **4.2** **91.76** **85** **100** **22.2**

**5** **3.2** **4.1** **89.35** **75** **95** **20.28** **12** **3.3** **4.2** **98.4** **65** **100** **22.86**

**6** **3.4** **4.15** **86.28** **60** **77.5** **18.9** **13** **3.3** **4.6** **92** **80** **195** **23.07**

**7** **3.6** **4.2** **84.1** **80** **100** **18.7** **14** **3.4** **4.3** **97.36** **75** **110** **22.12**

;

run;

**Proc** **discrim** data=d0805 testdata=t\_d pool=yes list crosslist listerr testlist out01 testout02 ;

Class group;

Var x1-x6;

priors '1'=**0.5** '2'=**0.5**;

**run**;

**Proc** **discrim** data=d0805 testdata=t\_d pool=no list crosslist listerr testlist testout03;

Class group;

Var x1-x6;

priors '1'=**0.5** '2'=**0.5**;

**run**;

**Proc** **discrim** data=d0805 testdata=t\_d pool=yes list crosslist listerr testlist testout04 ;

Class group ;

Var x1-x6;

priors '1'=**0.8** '2'=**0.2**;

**run**;

**四、实验实训结果与形式**

**1.实现任务的相关代码:**

**2.源代码截图:**

**3.运行结果截图：**

**五、实验实训素材（可另附文件）**