

**数 据 获 取 与 可 视 化**

**实 验 指 导 书**

**专业：计算机科学与技术**

**《数据获取与可视化》实验指导书**

**课程编码：6063122233 课程类型：课内实践教学**

**学时：16 学分：2**

**适用专业：工科类本科专业 开课学期：7**

**编写执笔人：陈 虹 审定负责人：刘红梅**

**必开实验个数：4**

一、课程简介

本课程是非独立设课，是计算机科学与技术专业本科生必修课《数据获取与可视化》的实验环节，是配合课堂教学的上机实验教学部分。通过一系列数据获取，数据采集与预处理，以及数据可视化的学习相应的实验练习，将数据获取与可视化的相关操作与理论知识融入实践中，加深学生对知识的理解。

二、课程教学目标

通过本课程的实验环节教学，深入理解python语言面向对象的概念、理论与技术；会编写网络爬虫代码进行数据采集并进一步预处理应用程序；掌握数据获取与可视化的新技术与新发展。通过实验，培养学生的计算思维和创新意识，提高学生的编程能力。

附、实验项目名称与学时分配

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 实验项目名称 | 类 型 | | | 学 时 |
| 验证性 | 综合性 | 设计性 |
| 1 | python基础与进阶知识 | √ |  |  | 4 |
| 2 | python爬虫以及数据存取操作 |  | √ |  | 2 |
| 3 | python数据清洗与数据预处理操作 |  | √ |  | 4 |
| 4 | 数据分析与可视化 |  |  | √ | 6 |
| 合计 |  |  |  |  | 16 |

1. 实验考核方式

实验成绩=实验报告成绩。共4份实验报告，每份100分，4份实验报告的均分为实验成绩。

1. 配套的实验教材及实验指导书

《Python数据分析与可视化：从入门到精通》高博、刘冰、李力编著，北京大学出版社

《Python数据分析与应用：从数据获取到可视化》黑马程序员编著，中国铁道出版社

《Python数据分析从入门到精通》 明日科技，贾晓红 编著 清华大学出版社

1. 实验项目

实验一、python基础与进阶知识 4课时

(一)实验类型：验证型

(二)实验类别：专业实验

(三)实验要求

(1)编写一个控制台应用程序，从键盘上输入球的半径，求球的表面积和体积。说明：球的表面积=4\*π\*r\*r；球的体积=4\*π\*r^3/3。

(2)通过对控制语句以及循环语句的学习能够实现随机数猜数 游戏。说明：可以选择难度，简单和困难模式直到猜中为止。

(四)实验目的

通过实验学会执行python命令以及脚本文件的方法；掌握python语言的数据类型、运算符、常量、变量、表达式和常用语句等基础知识；掌握程序的分支结构、循环结构、异常处理与程序调试；学会声明和调用函数的方法；理解掌握要求学生能够运用所学课程理论知识进行程序设计，培养其规范的编程习惯。

(五)实验内容

实验要求1：

1、实验代码

import math

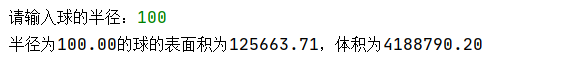
r = float(input("请输入球的半径："))

area = 4 \* math.pi \* r \* r

vol = 4 \* math.pi \* r \*\* 3 / 3

print("半径为%.2f的球的表面积为%.2f，体积为%.2f" % (r, area, vol))

1. 实验结果



实验要求2：

1、实验代码

import random

difficulty\_level = input("难度等级:(简单和困难)")

if difficulty\_level == "简单": # 判断选择的难度等级

randon\_number = random.randint(1, 100) # 随机数范围

number\_times = 0 # 次数

while True:

number = int(input("请输入一个数:"))

number\_times += 1

if number < randon\_number:

print("猜小了")

elif number > randon\_number:

print("猜大了")

else:

print("恭喜你,猜对了!你一共猜了{}次".format(number\_times))

break

if number\_times > 10:

print("少年还需努力")

elif difficulty\_level == "困难":

randon\_number = random.randint(1, 100) # 随机数范围

number\_times = 0 # 次数

chance = int(input("请输入你理想猜测的次数:"))

while True:

number = int(input("请输入一个数:"))

number\_times += 1

if number > randon\_number:

print("猜大了,你还有{}次机会".format(chance - number\_times))

elif number < randon\_number:

print("猜小了,你还有{}次机会".format(chance - number\_times))

else:

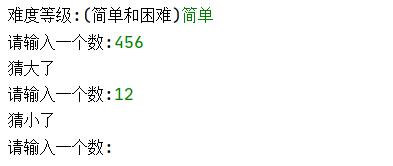
print("猜对了,真聪明!你一共猜了{}次".format(number\_times))

break

if chance - number\_times == 0:

print("给你机会你不中用啊")

2、实验结果



实验二、python爬虫以及数据存取操作 2课时

(一)实验类型：验证型

(二)实验类别：专业实验

(三)实验要求

(1)理解网络爬虫算法的原理，并设计使用Python语言获取网页数据的程序。

(2)爬虫获取百度首页信息。

(四)实验目的

强化Python程序的设计和编程能力；学习网络爬虫算法的原理；学习使用Python语言实现网络爬虫算法

(五)实验内容

1、实验代码

import requests #导入requests库

html\_url = ('https://baidu.com') #指定要爬取的url

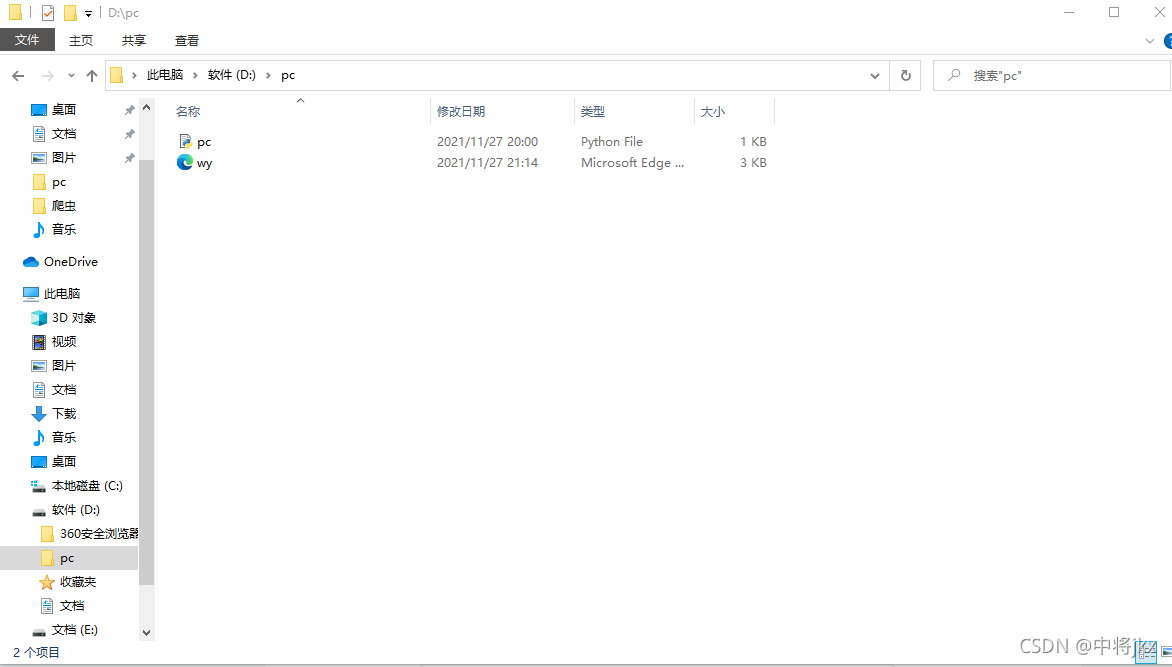
response = requests.get(html\_url) #发送get请求

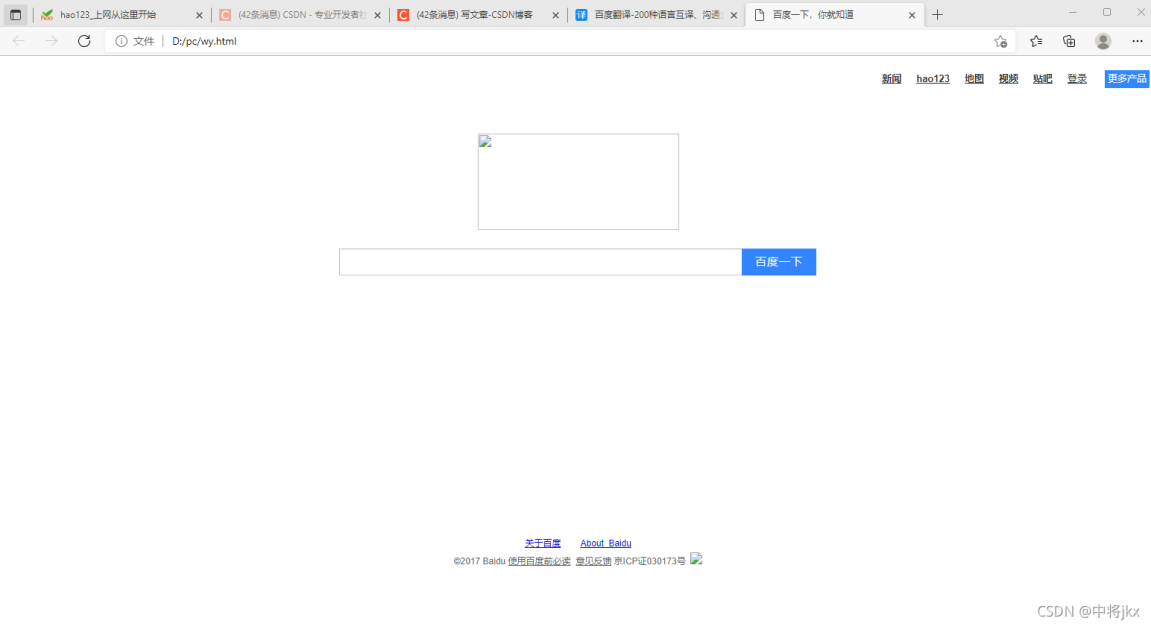
with open('D:\pc/wy.html','wb') as f:

f.write(response.content) #将爬取到的数据储存到D盘的pc文件夹

print('完成爬取！！！')

2、实验结果





实验三、python数据清洗与数据预处理操作 4课时

(一)实验类型：验证型

(二)实验类别：专业实验

(三)实验要求

利用pandas实现数据重复值。

(四)实验目的

数据重复会导致数据的方差变现，数据分布发生比较大变化。缺失会导致样本信息减少，不仅增加了数据分析的难度，而且会导致数据分析的结果产生偏差。异常值则会产生‘伪回归’。因此需要对数据进行检测，查询是否有重复值、缺失值和异常值，并且对所发现的异常数据进行适当的调整。

(五)实验内容

1、实验代码

import pandas as pd

detail = pd.read\_csv('data/detail.csv',index\_col=0,encoding='gbk')

# 定义去重函数

def delRep(list1):

list2=[]

for i in list1:

if i not in list2:

list2.append(i)

return list2

# 去重

# 提取dishes\_name所有数据转化为list

dishes = list(detail['dishes\_name'])

print('去重之前的所有菜品总数为：',len(dishes))

dish = delRep(dishes)

print('去重之后的所有菜品总数为：',len(dish))

去重之前的所有菜品总数为： 10037

去重之后的所有菜品总数为： 145

# 利用set方法去重

print('去重之前的所有菜品总数为：',len(dishes))

dish\_set = set(dishes)

print('去重之后的所有菜品总数为：',len(dish\_set))

去重之前的所有菜品总数为： 10037

去重之后的所有菜品总数为： 145

# 对dashes\_name去重

dishes\_name = detail['dishes\_name'].drop\_duplicates()

print('去重之后的所有菜品总数为',len(dishes\_name))

去重之后的所有菜品总数为： 145

#多列去重

print('去重之前订单详情表的形状为：',detail.shape)

shapeDet = detail.drop\_duplicates(subset=['order\_id','emp\_id'])

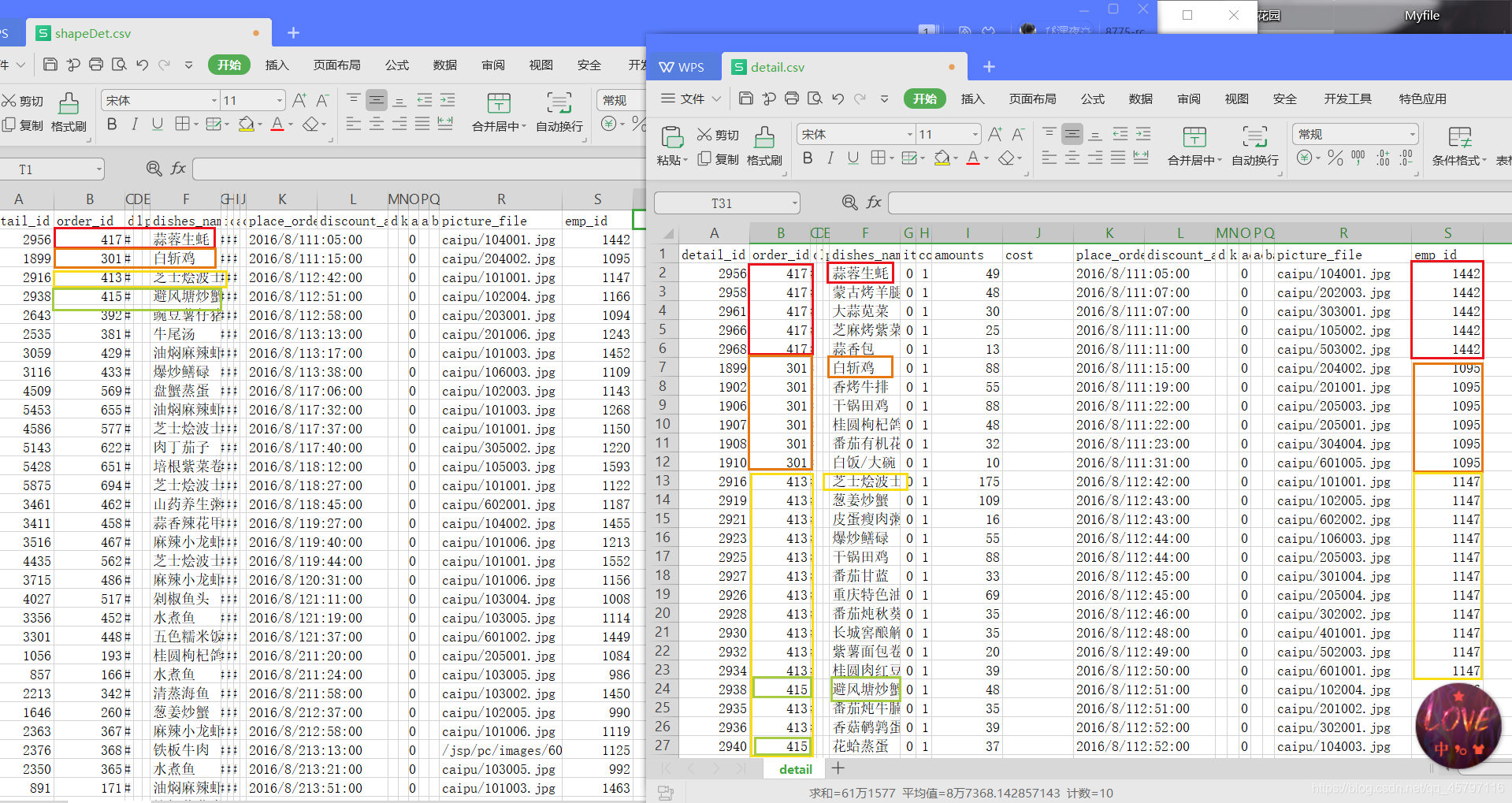
print('去重之后订单详情表的形状为：',shapeDet.shape)

shapeDet.to\_csv('data/shapeDet.csv',sep=',',index=True)

去重之前订单详情表的形状为： (10037, 18)

去重之后订单详情表的形状为： (942, 18)

2、实验结果



实验四、数据分析与可视化 6课时

(一)实验类型：验证型

(二)实验类别：专业实验

(三)实验要求

Matplotlib 是 Python 的绘图库，它能让使用者很轻松地将数据图形化，并且提供多样化的输出格式。可以用来绘制各种静态，动态，交互式的图表。是一个非常强大的 Python 画图工具，我们可以使用该工具将很多数据通过图表的形式更直观的呈现出来。本实验绘制直方图、折线图、散点图、饼图(选做)。

(1)根据

(2)

(3)

(四)实验目的

掌握Matplotlib库中的pyplot绘图模块，掌握pyplot模块中包含的一系列绘图函数的相关函数。

(五)实验内容

实验要求1：

1、实验代码

import numpy as np

import matplotlib.pyplot as plt

n\_groups = 5

class1 = (10, 15,20, 25, 30)

class2 = (30, 35, 40, 45, 50)

class3 = (50, 55, 60,65, 70)

fig, ax = plt.subplots()

index = np.arange(n\_groups)

bar\_width = 0.2

opacity = 0.5

error\_config = {'ecolor': '0.3'}

rects1 = ax.bar(index, class1, bar\_width,

alpha=opacity, color='indianred',

error\_kw=error\_config,

label='class 1')

rects2 = ax.bar(index + bar\_width, class2, bar\_width,

alpha=opacity, color='#106D9C',

error\_kw=error\_config,

label='class 2')

rects3 = ax.bar(index + bar\_width + bar\_width, class3, bar\_width,

alpha=opacity, color='#00A2DE',

error\_kw=error\_config,

label='class 3')

ax.set\_xticks(index + 3 \* bar\_width / 3)

ax.set\_xticklabels(('1', '2', '3', '4', '5'))

ax.legend()

plt.xlabel(u"X")

plt.ylabel(u'Count')

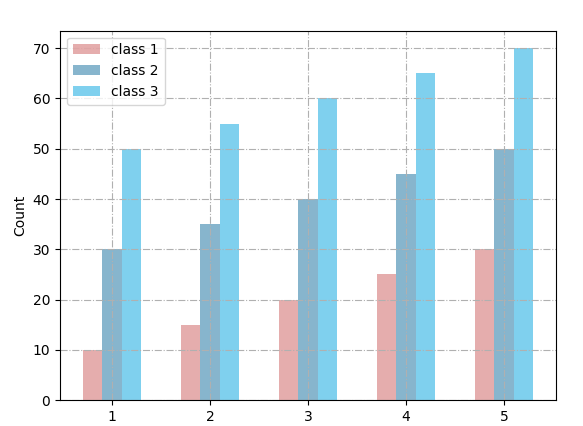
fig.tight\_layout()

# plt.savefig('result.png', dpi=200)

plt.grid(ls='-.') # 绘制背景线

plt.show()

2、实验结果



实验要求2：

1、实验代码

import matplotlib.pyplot as plt

import numpy as np

x = [10, 20, 30, 40, 50]

y1 = [4, 4, 5, 2, 1]

y2 = [3, 2, 1, 4, 2]

y3 = [2, 1, 4, 5, 3]

y4 = [1, 3, 2, 3, 4]

#绘制折线图，添加数据点，设置点的大小

# 此处也可以不设置线条颜色，matplotlib会自动为线条添加不同的颜色

plt.plot(x, y1, alpha=1, color="c", marker='\*', markersize=7)

plt.plot(x, y2, alpha=1, color='b', marker='x',markersize=7)

plt.plot(x, y3, alpha=1, color='y', marker='o',markersize=7)

plt.plot(x, y4, alpha=1, color='r', marker='^',markersize=7)

# plt.yticks(np.arange(0.5, 3.25, step=0.25))

# plt.ylim(ymin=-0.005)

# plt.xticks(np.arange(8, 70, step=8))

plt.title('title') # 折线图标题

plt.xlabel('X') # x轴标题

plt.ylabel('Y') # y轴标题

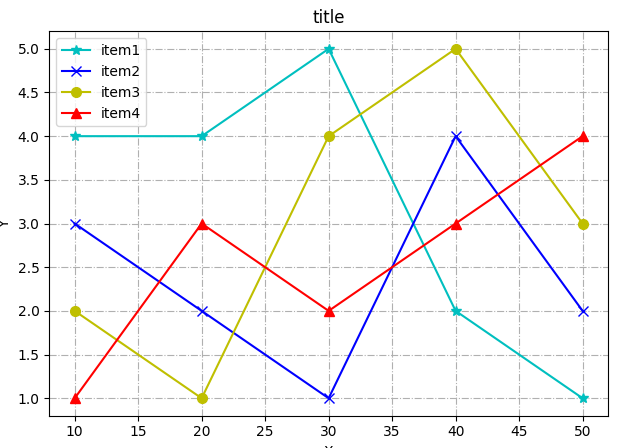
plt.grid(ls='-.') # 绘制背景线

plt.legend(['item1', 'item2', 'item3', 'item4'])

plt.tight\_layout()

plt.show()

2、实验结果



实验要求3:

1、实验代码

from numpy.random import randn

import matplotlib.pyplot as plt

plt.style.use('ggplot')

plot\_data1 = randn(50).cumsum()

plot\_data2 = randn(50).cumsum()

plot\_data3 = randn(50).cumsum()

plot\_data4 = randn(50).cumsum()

fig = plt.figure()

ax1 = fig.add\_subplot(1,1,1)

ax1.plot(plot\_data1, marker=r'o', color=u'blue', linestyle='-', label='Blue Solid')

ax1.plot(plot\_data2, marker=r'+', color=u'red', linestyle='--', label='Red Dashed')

ax1.plot(plot\_data3, marker=r'\*', color=u'green', linestyle='-.', label='Green Dash Dot')

ax1.plot(plot\_data4, marker=r's', color=u'orange', linestyle=':', label='Orange Dotted')

ax1.xaxis.set\_ticks\_position('bottom')

ax1.yaxis.set\_ticks\_position('left')

ax1.set\_title('Line Plots: Markers, Colors, and Linestyles')

plt.xlabel('Draw')

plt.ylabel('Random Number')

plt.legend(loc='best')

plt.savefig('line\_plot.png', dpi=400, bbox\_inches='tight')

plt.show()

2、实验结果



实验要求4（选做）:

1. 实验代码

from matplotlib import pyplot as plt

from matplotlib import font\_manager

my\_font=font\_manager.FontProperties(fname="C:/Windows /Fonts/STFANGSO.TTF",size=14)

x=[15,30,45,10]

label\_list=["Frogs","Hogs","Dogs","Logs"]

color=["blue","orange","green","red"]

explode=[0,0.09,0,0]

plt.figure(figsize=(15,15),dpi=100)

patches,l\_text,p\_text=plt.pie(x,

explode=explode,

colors=color,

labels=label\_list,

labeldistance=1.1,

autopct="%1.1f%%",

shadow=True,

startangle=90,

pctdistance=0.6

)

plt.legend(prop=my\_font,loc="upper right")

plt.show()

1. 实验结果

