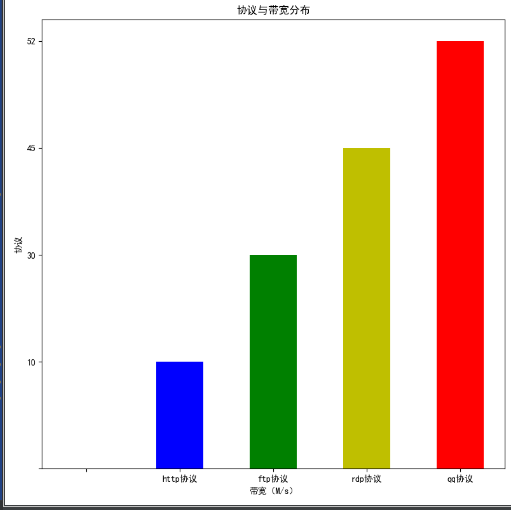
**Python经典网络协议 第六天作业**

**1、matplotlib柱状图**

代码：

from matplotlib import pyplot as plt  
  
# 设置成中文  
plt.rcParams['font.sans-serif'] = ['SimHei']  
plt.rcParams['font.family'] = 'sans-serif'  
colorlist = ['r', 'b', 'g', 'y']  
  
  
def mat\_zhu(name\_list, size\_list):  
 # 调节图形大小，宽，高  
 plt.figure(figsize=(8, 8))  
 # 横向柱状图  
 # plt.barh(name\_list, size\_list, height=0, color=colorlist)  
 # 竖向柱状图  
 plt.bar(name\_list,sorted(size\_list),width=0.5, color=colorlist)  
 # 添加主题和注释  
 plt.title('协议与带宽分布') # 主题  
 plt.xlabel('带宽（M/s）') # x轴注释  
 plt.ylabel('协议') # Y轴注释  
 # 保存图片  
 plt.savefig('result1.png')  
 plt.show()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 size = ['','30','52','10','45']  
 name = ['','http协议','ftp协议','rdp协议','qq协议']  
 mat\_zhu(name,size)

运行结果：



**2.matplotlib线性图**

代码：

from matplotlib import pyplot as plt  
import random  
import datetime  
  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文  
plt.rcParams['font.family'] = 'sans-serif'  
  
  
def mat\_line(cpu\_usage\_list):  
 # 调节图形大小，宽，高  
 fig = plt.figure(figsize=(6, 6))  
 # 一共一行, 每行一图, 第一图  
 ax = fig.add\_subplot(111)  
  
 # 处理X轴时间格式  
 import matplotlib.dates as mdate  
 # ax.xaxis.set\_major\_formatter(mdate.DateFormatter('%Y-%m-%d %H:%M:%S')) # 设置时间标签显示格式  
 ax.xaxis.set\_major\_formatter(mdate.DateFormatter('%H:%M')) # 设置时间标签显示格式  
  
 # 处理Y轴百分比格式  
 import matplotlib.ticker as mtick  
 ax.yaxis.set\_major\_formatter(mtick.FormatStrFormatter('%d%%'))  
  
 # 把cpu\_usage\_list的数据,拆分为x轴的时间,与y轴的利用率  
 x = []  
 y = []  
  
 for time, cpu in cpu\_usage\_list.items():  
 x.append(time)  
 y.append(cpu)  
  
 # 添加主题和注释  
 plt.title('路由器CPU利用率')  
 plt.xlabel('采集时间')  
 plt.ylabel('CPU利用率')  
  
 fig.autofmt\_xdate() # 当x轴太拥挤的时候可以让他自适应  
  
 # 实线红色  
 ax.plot(x, y, linestyle='solid', color='r', label='R1')  
 # 虚线黑色  
 # ax.plot(x, y, linestyle='dashed', color='b', label='R1')  
  
 # 如果你有两套数据,完全可以在一幅图中绘制双线  
 # ax.plot(x2, y2, linestyle='dashed', color='b', label='R2')  
  
 # 设置说明的位置  
 ax.legend(loc='upper left')  
  
 # 保存到图片  
 plt.savefig('result1.png')  
 # 绘制图形  
 plt.show()  
if \_\_name\_\_ == '\_\_main\_\_':  
 now = datetime.datetime.now()  
 now\_t3 = (datetime.datetime.now() - datetime.timedelta(hours=3))  
 now\_t6 = (datetime.datetime.now() - datetime.timedelta(hours=6))  
 now\_t9 = (datetime.datetime.now() - datetime.timedelta(hours=9))  
 now\_t12 = (datetime.datetime.now() - datetime.timedelta(hours=12))  
 now\_t33 = (datetime.datetime.now() + datetime.timedelta(hours=3))  
 now\_t66 = (datetime.datetime.now() + datetime.timedelta(hours=6))  
 now\_t99 = (datetime.datetime.now() + datetime.timedelta(hours=9))  
 now\_t1212 = (datetime.datetime.now() + datetime.timedelta(hours=12))  
 cpu\_list = {now\_t12: random.randrange(0,100),  
 now\_t9: random.randrange(0,100),  
 now\_t6: random.randrange(0,100),  
 now\_t3: random.randrange(0,100),  
 now: random.randrange(0,100),  
 now\_t33: random.randrange(0,100),  
 now\_t66: random.randrange(0,100),  
 now\_t99: random.randrange(0,100),  
 now\_t1212: random.randrange(0,100)}  
 mat\_line(cpu\_list)

运行结果：

