可视化测试三

（时间限制: 2 小时， 总分: 150 分）

## 做题要求（重点阅读）：

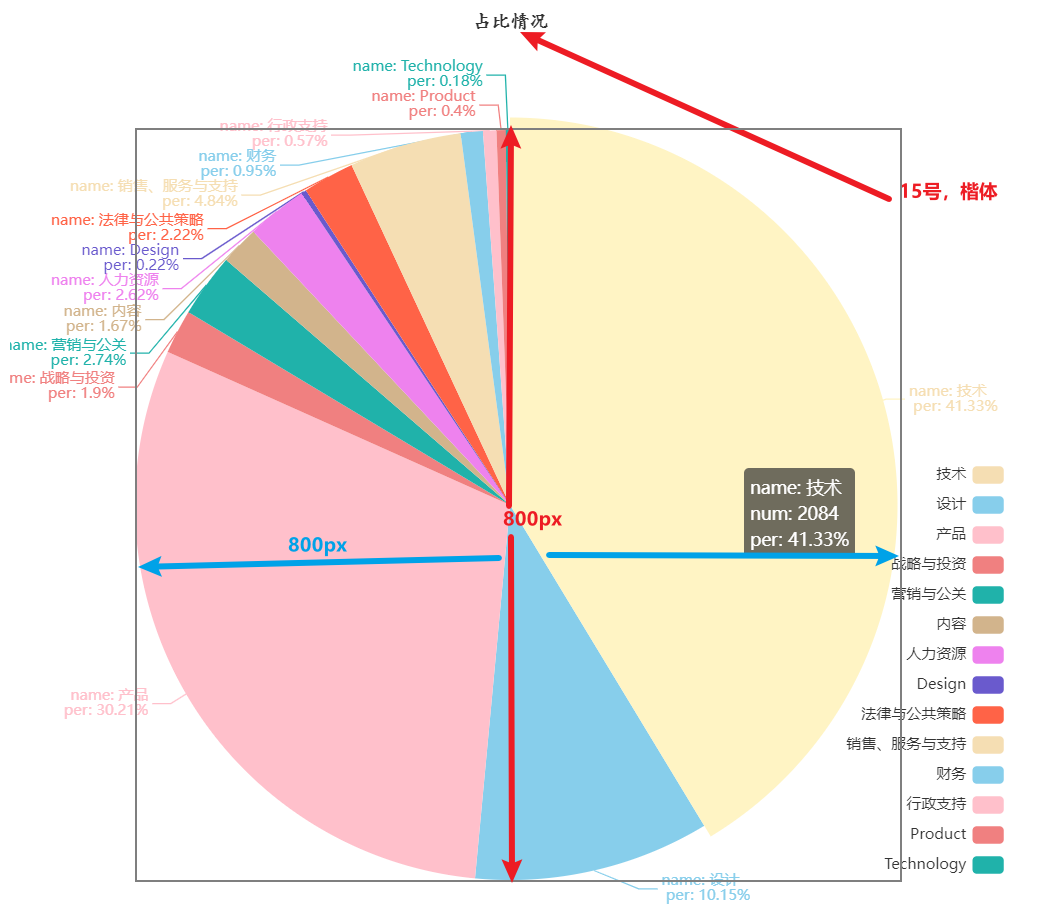
出现阅读问题，概不解释！！！！

1. **一定要仔细读题！！！！！**
2. **写答案的时候，如果有多个答案时，应该都写出来；**
3. **答案一定要写全，不写全，按照写的内容给分；**
4. **写代码的时候，要标清写在哪个文件里面；**
5. **HTML 文件中，只需要写出我要求的信息，文件头不需要考虑。**

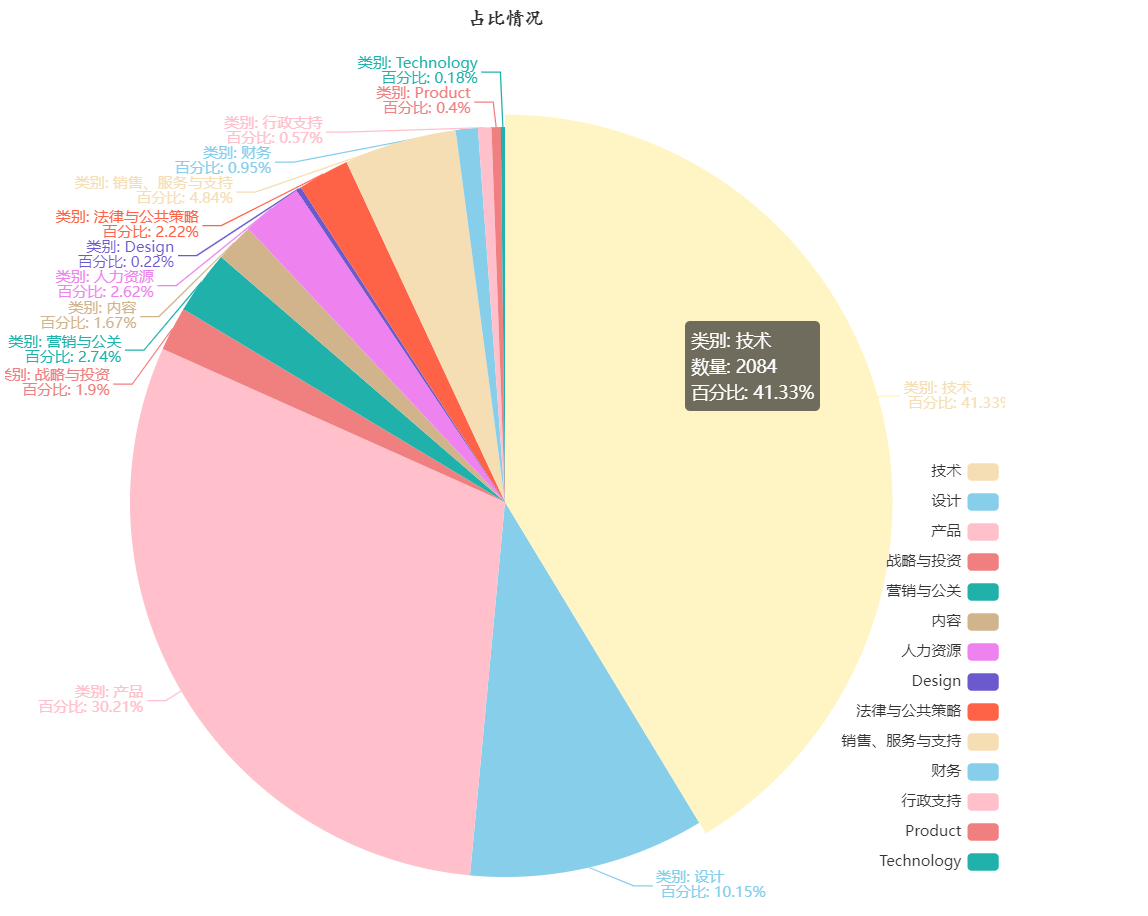
## 题目要求（提出问题）：

**小明前两天利用爬虫爬取了腾讯招聘这个网站，获取了一些自己想要分析的数据，并存到了数据库里面，但是他不知道从来都没有学过数据处理以及可视化相关的技术，所以，他广发英雄帖，号召天下豪杰来帮他解决以下几个问题：**

1. **求出各个类别岗位需求总数，展示出各类别岗位需求总数的占比情况，并分析出该图形所要表达的含义，图表要求如下：**
   * **图表各区域的颜色分别为：['wheat', 'skyblue', 'pink', 'lightcoral', 'lightseagreen', 'tan', 'violet', 'slateblue', 'tomato']；**
   * **具体情况如图所示：（图1-1是标识， 按照图 1-2来做）**

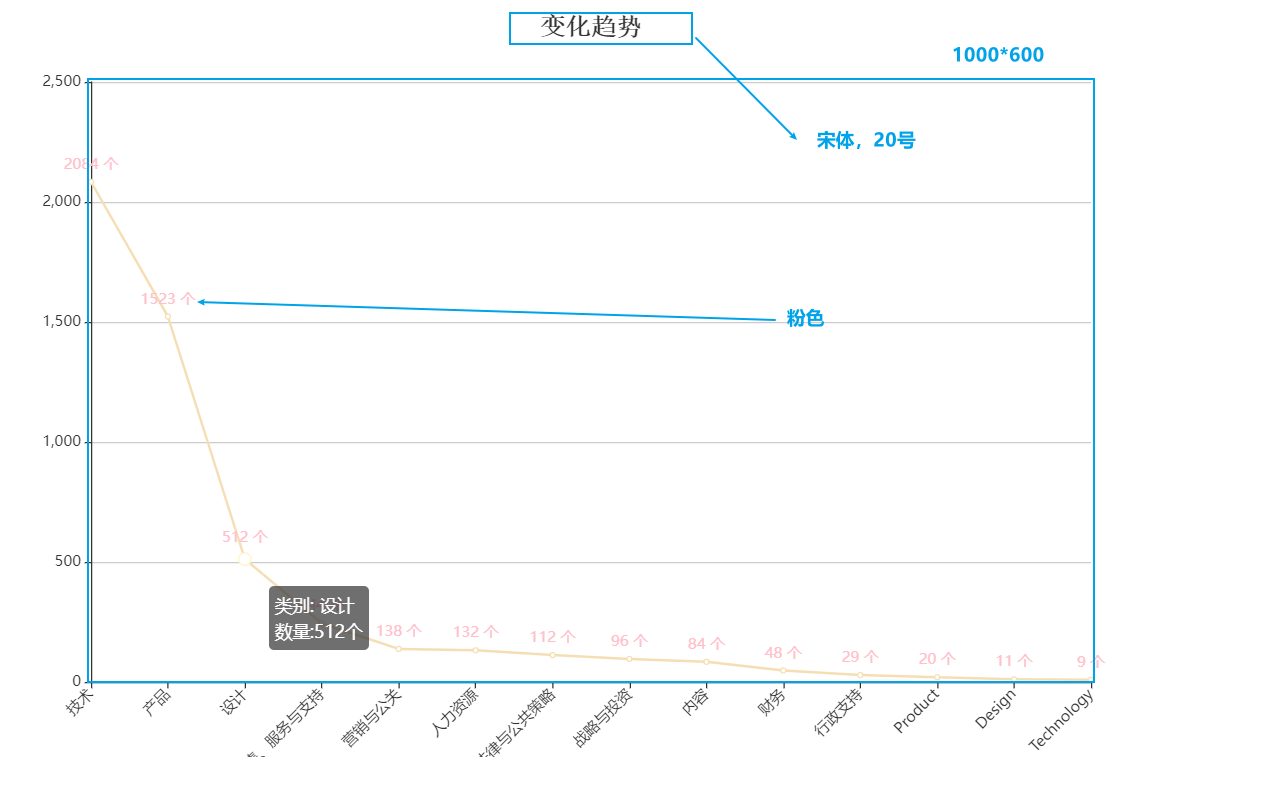


**图1-1**



**图1-2**

1. **求出各个类别岗位需求总数，按照岗位需求总数降序排序，展示出各类别岗位需求总数的变化趋势，图表要求如图所示：**



**图2-1**

## 数据相关（数据获取、 分析问题）：（15 分）

**你收到了小明发的英雄帖，你经过一番分析之后，还是决定先看看数据之后再做定夺，你向小明提出了自己需求，小明告诉你数据就存在电脑上的数据库（test）里面，并告诉你本机的 ip 地址是 172.168.5.223 、数据库的登录密码是 root 以及端口是 3309，于是，你利用自己已经学会的数据库操作，开始大展拳脚：**

1. **登录数据库：**

**命令：mysql -uroot -proot -h172.168.5.223 -P3309**

1. **查看有哪些数据库：**

**命令：show databases;**

**输入命令后，如图所示：**

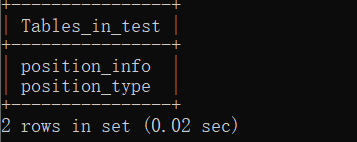


1. **查看有哪些表：**

**命令：use test;**

**show tables;**

**输入命令后，如图所示：**

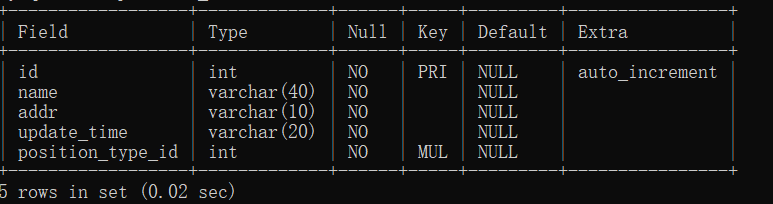


1. **查看表结构：(从上往下的顺序查看)**

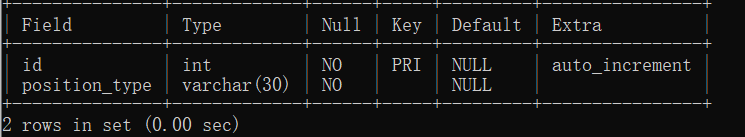
**命令：desc position\_info;**

**desc position\_type;**

**输入命令后，如图4)-1、图4)-2所示：**



**图 4)-1**

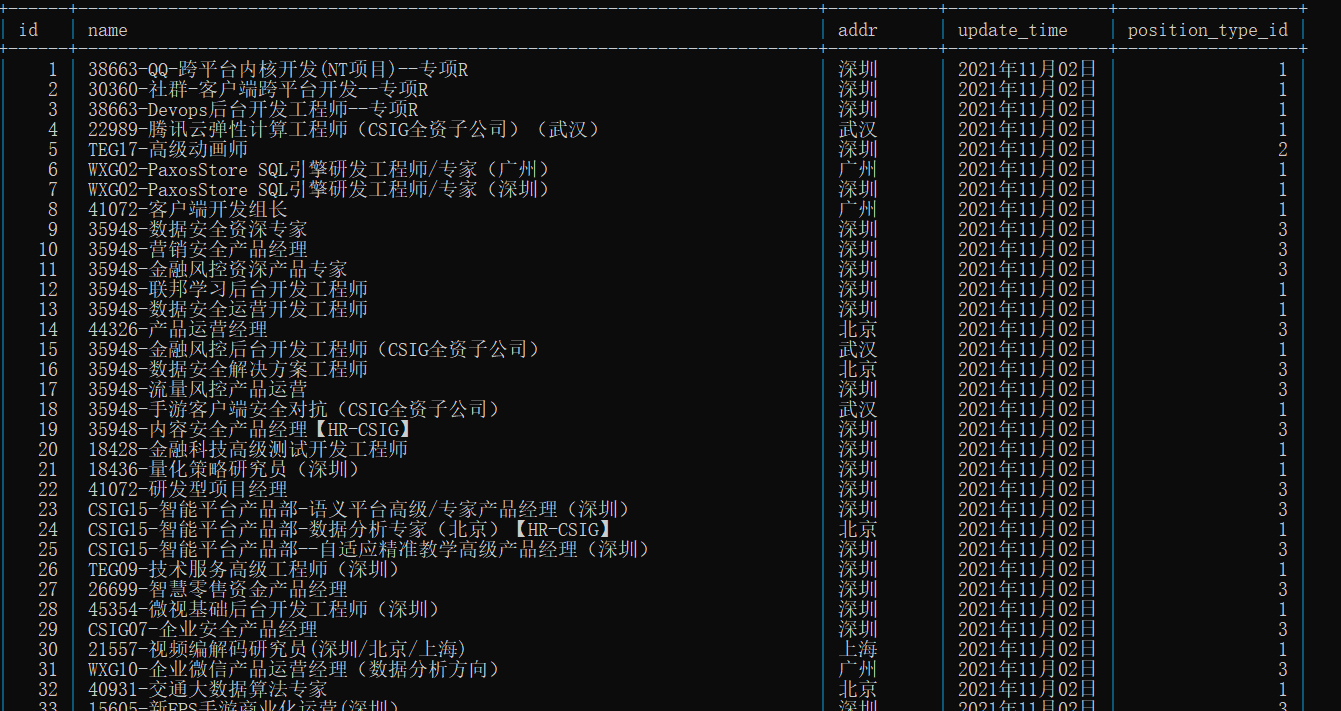


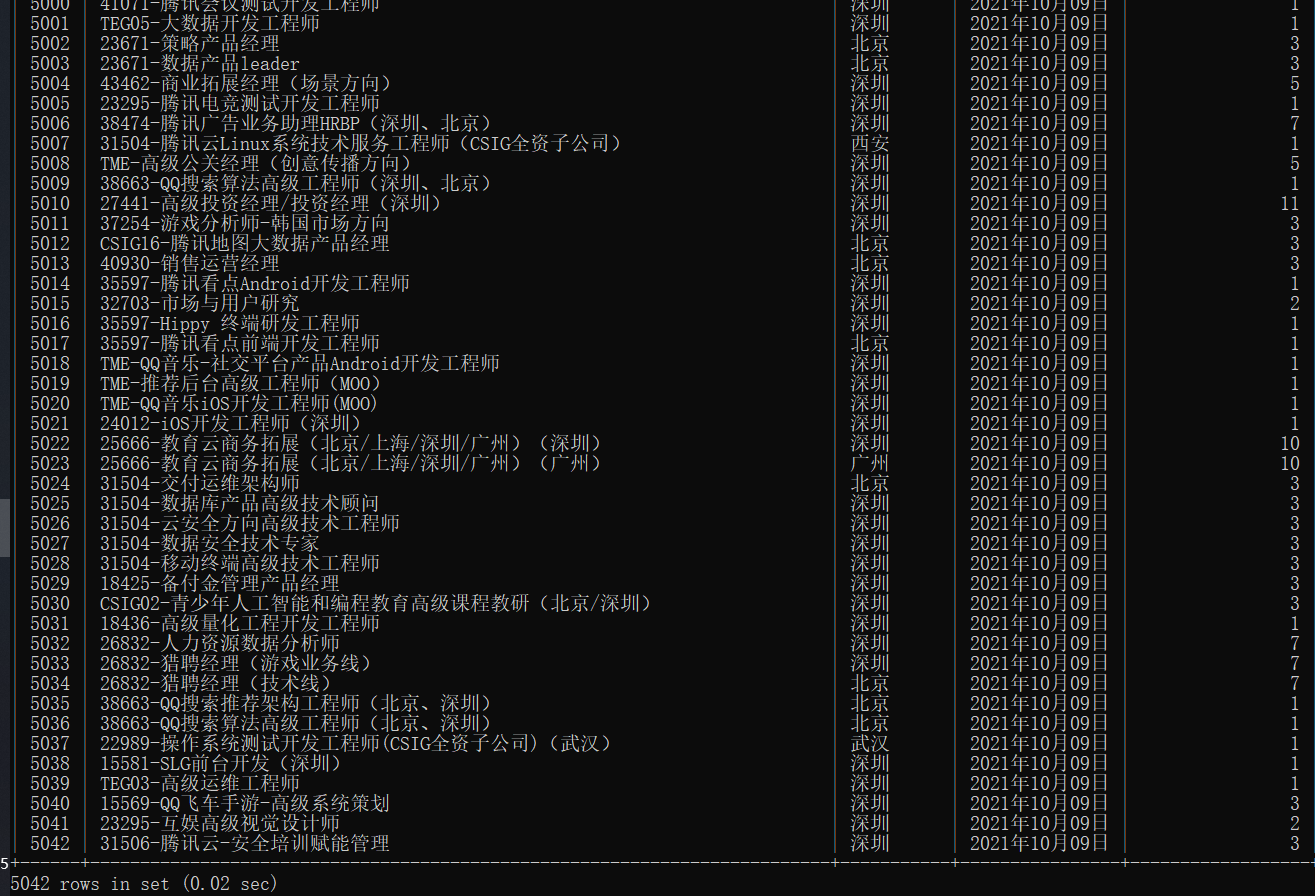
**图4)-2**

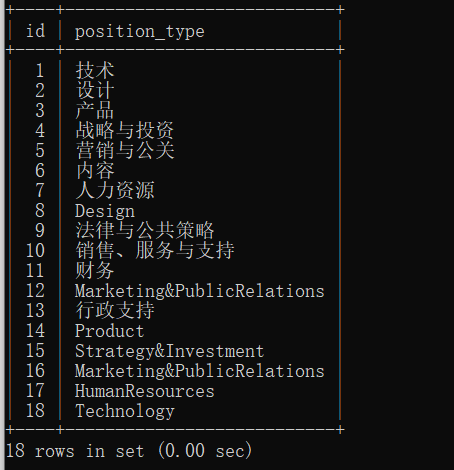
1. **查看表数据：**

**命令：select \* from position\_info;**

**select \* from position \_type;**







## 做题思路（制定计划）：（25 分）

**你看了表数据之后，对题目进行了下一步分析，归纳了一些要点，并把思路告诉了小明：**

**你的思路：**

* **1. 类别， 类别总数；**
* **首先，从 position\_type 取出题目所需类别，然后，再求出 position\_info 中各岗位的总数，将其渲染成饼图。**
* **1. 类别，类别总数，降序。**
* **首先，从 position\_type 取出题目所需类别，然后，再求出 position\_info 中各岗位的总数并按照岗位总数降序排序。将其渲染成折线图。**

## 解题过程（数据处理、 实施计划）：（90分）

**你把思路告诉小明之后，小明觉得你说的内容完全符合他的要求，并让你全权负责处理这件事情，于是，你又对题目进行了下一步的分析：**

**首先创建了一个项目文件夹，命名为 Visualization\_test3，你在项目文件夹下面分别创建了 templates（模板文件夹）、static（静态文件夹）、views（子路由相关的文件）、config（连接数据的配置文件夹）、databases （数据库文件夹）等对应的directory / package，除此之外，你在项目文件夹下又创建了一个 app.py 文件, 在static 文件夹下面复制了一个 echarts.min.js 文件，在 templates 下面创建了 test\_pie.html和 test\_line.html文件；config 文件夹下面有一个 \_\_init\_\_.py 文件；databases 下面有 position\_info.py 和 position\_type.py文件；views（PS： 如果没学到蓝图(blueprint)，这里就不用管）下面有 \_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_文件。**

1. **通过sqlalchemy 取出所需数据：（30分）**
   1. **导入库：（\_\_init\_\_.py）**
      1. **from sqlalchemy import create\_engine**
      2. **from sqlalchemy.orm import sessionmaker**
      3. **from sqlalchemy.ext.decalative import decalative\_base**
   2. **连接数据库：**
      1. **engine = create\_engine(‘mysql+pymysql://root:root@172.168.5.223:3309/test?charset=utf8’)**
      2. **Base = decalative\_base()**
      3. **Session = sessionmaker(bind=engine)**
   3. **创建表映射关系：**
      1. **position\_info.py**
      2. **class posion(Base):**
         1. **\_\_tablename\_\_=’position\_info’**
         2. **id = Column(Integer, primary\_key=True, auto)**
         3. **name=Column(String(), nullable=False)**
         4. **addr = Column(String(10), nullable=False)**
         5. **update\_time = Column(String(20), nullable=False)**
         6. **position\_type\_id=Column(Integer, ForeignKey('position\_type.id'))**
      3. **position\_type.py**
      4. **class type(Base):**
         1. **\_\_tablename\_\_ = 'position\_type'**
         2. **id = Column(Integer, primary\_key=True)**
         3. **position\_type = Column(String(30), nullable=True)**
   4. **取出所需数据：**
      1. **position\_info = session.query(position.position\_type\_id,func.count(position.id)).group\_by(position.position\_type\_id).all()**
         1. **(1, 2084)**
   5. **处理数据：**
      1. **for item in position\_info:**
         1. **for info in session.query(position\_type).filter(item[0] == position\_type.position\_id):**
2. **通过 Flask 进行模板渲染：（24分）**
   1. **导入库：**
      1. **from flask import Flask, render\_template**
   2. **初始化对象：**
      1. **app = Flask(\_\_name\_\_)**
   3. **指定路由并创建视图函数（视图函数中模板写出其对应关系）：**
      1. **@app.route(‘/pie’)**
      2. **def position\_pie():**

**return render\_template(‘test\_pie.html’, data=data\_pie )**

* 1. **启动节点，并开启调试模式（要求：必须写main 函数）**
     1. **if ‘\_\_name\_\_’==’\_\_main\_\_’:**

**app.run(debug=True)**

1. **用 echarts.js 进行页面图表展示：（36分）**
   1. **导入插件：**
      1. **<script src=’../static/echarts.min.js’></script>**
   2. **定义一个盒子：**
      1. **<div style=””></div>**
   3. **获取盒子的 document 对象：**
      1. **var doc\_div = document.querySelector(‘div’);**
   4. **把获取的document对象实例化到echarts 中：**
      1. **var myChart = echarts.init(doc\_div);**
   5. **编写配置项：**
      1. **var option = {**
         1. **title: {},**
         2. **legend: {}，**
      2. **}**
   6. **实例化配置项：**
      1. **myChart.setOption(option);**

## 分析文档（解决问题）：（20 分）

**你经过一番猛如虎的操作之后，终于实现了小明的要求，并告诉他在/pie路径下是占比情况，在\_\_/line\_路径下是变化趋势，小明没听懂，让你操作一下，于是你运行代码，在浏览器中输入了\_\_\_\_\_\_\_\_\_http://127.0.0.1:5000/pie\_\_\_\_\_\_\_\_\_\_\_\_\_\_，展示了如 1\_1 所示的图表，又在浏览器输入了\_\_\_\_\_\_\_\_\_\_\_\_\_ http://127.0.0.1:5000/line \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，展示了如 1\_2 所示的图表，小明恍然大悟，让你对这两张图进行分析，告诉他这两张图分别表示什么含义：**