HW\_Week2\_Sat\_Thinking

Xiang Guo

**Thinking 1 MVC框架指的是什么**

MVC是一种模式的首字母缩写，代表 Model-View-Controller（模型-视图-控制器） 模式。这种模式用于web应用程序的分层开发。

模型

=====

就是业务流程/状态的处理以及业务规则的制定。代表一个存取数据的对象，也可以带有逻辑。业务流程的处理过程对其它层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是MVC最主要的核心。

一般包括包括Service层和Dao层；

Service层用于和数据库联动，放置业务逻辑代码，处理数据库的增删改查，

Dao层用于放各种接口，以备前端调用.

视图

=====

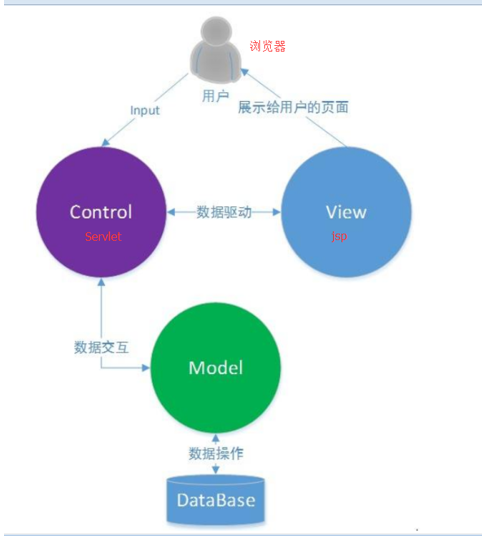
是Web应用中处理响应给客户的页面的部分，也就是用户交互界面。例如我们写的html静态页面，jsp动态页面，这些最终响应给浏览器的页面都是视图. 一个应用可能有很多不同的视图，MVC设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。

控制

=====

控制(Controller)可以理解为从用户接收请求, 将模型与视图匹配在一起，共同完成用户的请求。它负责根据用户从"视图层"输入的指令，选取"数据层"中的数据，然后对其进行相应的操作，产生最终结果。

以下是在网上找到的一张图：



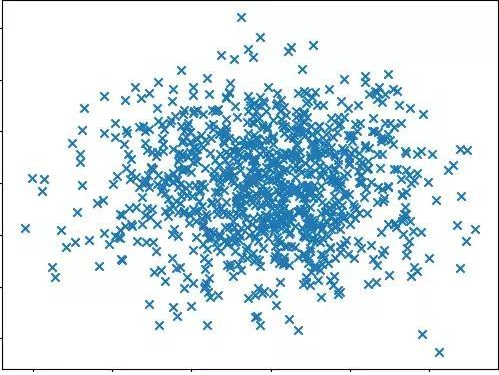
**Thinking 2 基于Python的可视化技术都有哪些，你使用过哪些**

Python 提供很多可视化的技术以及相应的库。课上主要讲了matplotlib和seaborn这两种库，都提供很多可视化试图来对数据进行绘制。我自己也是使用matplotlib比较多，主要是使用一些静态的可视化视图。

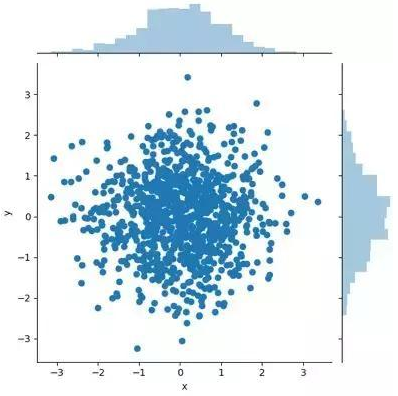
**散点图**

首先是可以绘制散点图，将两个变量的值显示在二维坐标里，适用于展示两个变量之间的关系。

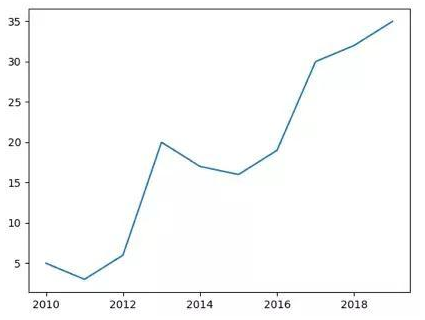
Matplotlib:



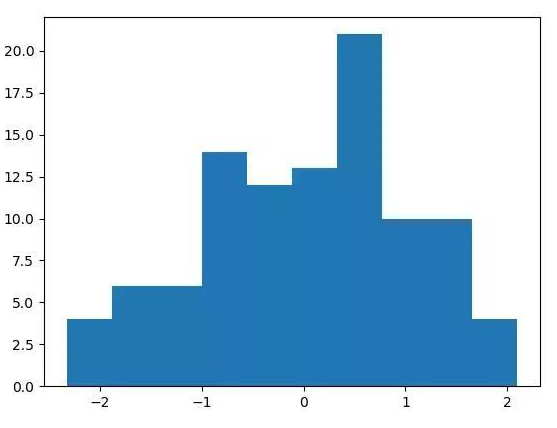
Seaborn:



**折线图 （更容易反映变化的趋势）**

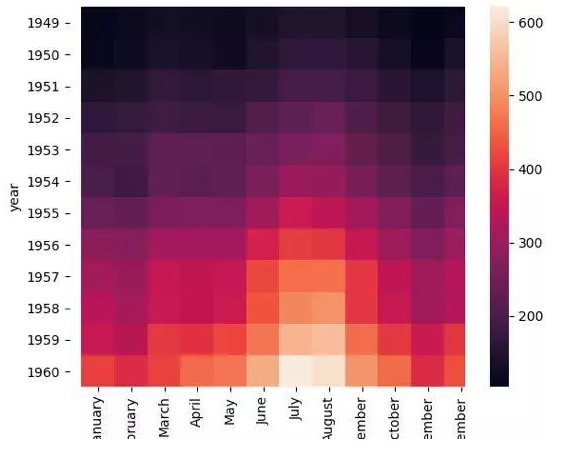


**直方图（反映了数据在不同横轴下的一个变化，容易对比）**



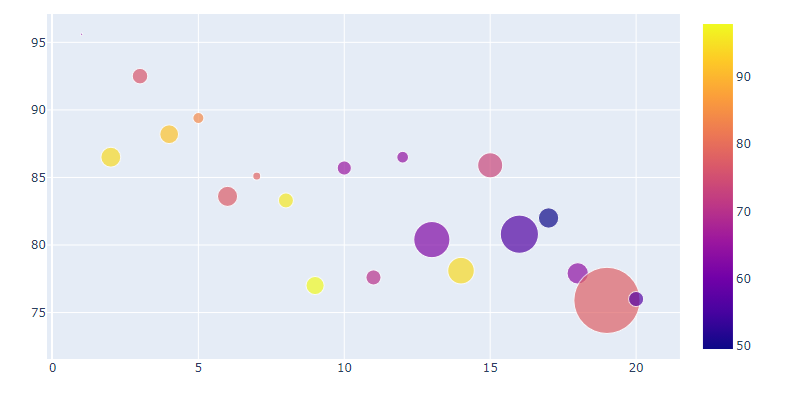
**热力图**

**是一种矩阵表示方法，其中矩阵中的元素值用颜色来代表，不同的颜色代表不同大小的值。通过颜色就能直观地知道某个位置上数值的大小**

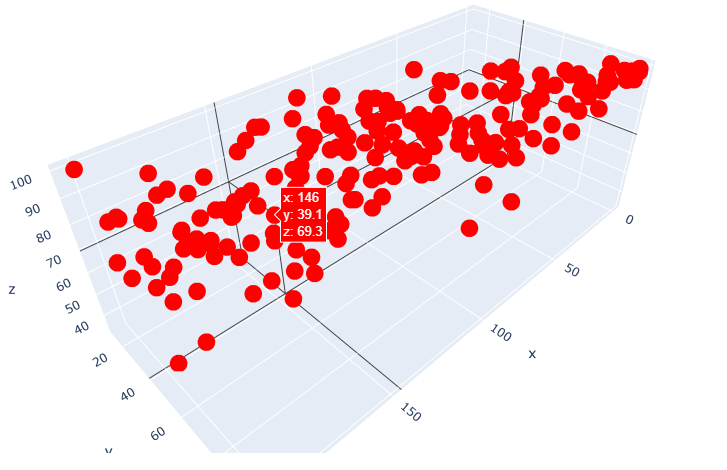


以上主要是静态的可视化视图。此外还有一些交互式可视化的技术比如梳理，过滤，缩放和悬停。Python有相应的可视化库Bokeh和Plot.ly来实现。

以下是kaggle上面教程中plot.ly的气泡图：



以及3D散点图，鼠标悬停之后可以看到数据：



这是Kaggle的教程地址：<https://www.kaggle.com/kanncaa1/plotly-tutorial-for-beginners>