

1 概述

1.1 项目结果预告



1.2 项目摘要

许多不健康的饮食方式已经渗透人们的生活，如果能了解怎样的饮食结构会导致一些疾病，就有可能提前预防。我们用数据可视化的形式，帮助人们了解良好的饮食结构，让人们能够直观探究饮食结构与患病概率的关系。我们的项目显示世界各地人民的饮食结构及其肥胖率与营养不良率，并尝试建模不同的饮食结构对患病几率的影响，允许用户通过调整饮食结构变量来估计患病的风险。

2 数据

2.1 数据描述

- 本数据集中共有五份数据，均为CSV类型数据，数据大小从4k到44k不等
- 五份数据的名称分别为
 - Fat_Supply_Quantity_Data.csv(42KB)
 - Food_Supply_kcal_Data.csv(43KB)
 - Food_Supply_Quantity_kg_Data.csv(44KB)
 - Protein_Supply_Quantity_Data.csv(43KB)
 - Supply_Food_Data_Descriptions.csv(4KB)
- 如上面表名顺序，首先介绍前四个表，因为这四个表中数据结构相同
 - 概述：这四个表分别计算了不同类别食物的脂肪量、能量摄入(kcal)、食物供应量(kg)和蛋白质(均以总摄入量的百分比计算)。其中还添加了肥胖和营养不良率(也是百分比)以进行比较。数据集的末尾还包括最新的新冠确诊/死亡/康复/活跃病例(也包括每个国家当前人口的百分比)。
 - 数据的每一列包含一个属性，分别是(酒精，肉类，鸡蛋，肥胖率，死亡率，康复率等)
 - 数据的每一行也包含一个属性，分别是统计到的170个国家的名称
 - 表中的每一个数值表达的是该数据所在行对应的国家在该数据所在列对应属性上的表现
- 不同食物组的供应量、营养价值、肥胖和营养不足百分比的数据来自联合国粮食及农业组织[粮农组织网站](#)要从粮农组织数据中查看每个类别中包含的具体食物类型，请查看最后一个数据集Supply_Food_Data_Descriptions.csv

2.2 数据来源

https://www.kaggle.com/mariaren/covid19-healthy-diet-dataset?select=Fat_Supply_Quantity_Data.csv

2.3 数据预处理

- 由于列属性太多，难以全部可视化，因此将类似的属性进行合并，并删除一些不需要的属性。同时有一些数据不符合可视化的要求和伦理常识，也进行删除
 - 合并属性：(蔬菜，水果) => 蔬果，(糖，糖类产品) => 糖，(动物产品，肉) => 动物产品，其他等
 - 删除属性：与新冠肺炎相关的属性如死亡率等；人口属性
 - 不符合可视化要求：部分数据是 NAN，删除这样的国家
 - 不符合伦理常识：将中国和台湾省分开，删除台湾省
- 最终呈现：
 - 共168行，24列，主要属性9个，分别是：酒精饮料，动物产品，谷物，鸡蛋，海鲜，蔬果，牛奶，糖，植物，其他（后面的数据除了肥胖率和营养不良率，都包含在其他中）
 - 程序：prepare.ipynb , combine.ipynb

3 目标和任务

1. 充分展现数据的所有特征，展现数据的潜在的多维价值

对于数据中食物的来源，对其分类，提供有用的分类信息；对于多个国家的不同饮食结构，对其地图可视化，将国家这一维数据拓展为二维空间数据，提供用户探索数据特征与空间位置的途径；将每一个国家的饮食结构可视化，突出主要矛盾。

2. 利用数据之间的交互关系，提供用户分析数据的边界工具

将繁琐的数据发掘拓展为图形化的交互方式，用户可以根据特殊需要更改探索的维度(因变量)，探究在不同因素引导之下的数据发展趋势，对于不同方向的研究课题具有启发意义。

3. 利用机器学习，基于现有数据预测饮食结构模型

不仅帮助用户提供现有数据特征的分析，提供一种有价值的思路：基于机器学习等模型探索在修改相应因素下对肥胖率等关键问题的解决方法，提供决策方向。

4 可视化界面

我们最终的可视化界面如下图所示。



4.1 视图①



图1

视图①为菜单栏，四个图形选项为摄入类型，将鼠标依次放置在相应图标会显示脂肪、食物、热量、蛋白质摄入量，如图 1 所示。数据来源中有以上四类，它们各自整体结构相似，设计菜单栏作为导航，用户可选择相应的摄入类型作为研究目标。视图交互方面，视图①与视图②的共同作用产生对视图③的交互；视图①与视图③的共同作用产生对视图④的交互，具体交互效果将在后续详细描述。菜单栏使用 echarts 库的 toolbox 编写，可以方便的更换按钮的矢量图标，同时因为与之交互的视图均用 echarts 编写，接口较为统一。交互事件为通过不同的按钮来更换后台读取的文件名，更新数据来源的同时刷新视图。

4.2 视图②



图2

视图②为旭日图，描述了数据的食物种类。中心的圆圈表明视图名称，食物种类分为九大类，依次对应每个扇形。在初始概览图中可以看到每一类的粗略描述，如图2所示。用户将鼠标移动到相应扇形的每一类都会高亮显示，选择具体类别后旭日图会展开，可以查看该类的具体描述，如图3所示，同时点击中心圆形会返回到上一层界面。实现方面采用echarts的'sunburst'，echarts在旭日图相比d3有更为完善和易用的动画过渡。用户点击相应扇形之后不仅可以查看对应类别的具体组成，还会选定相应的研究对象，具体产生的交互结果将展示在视图③中。

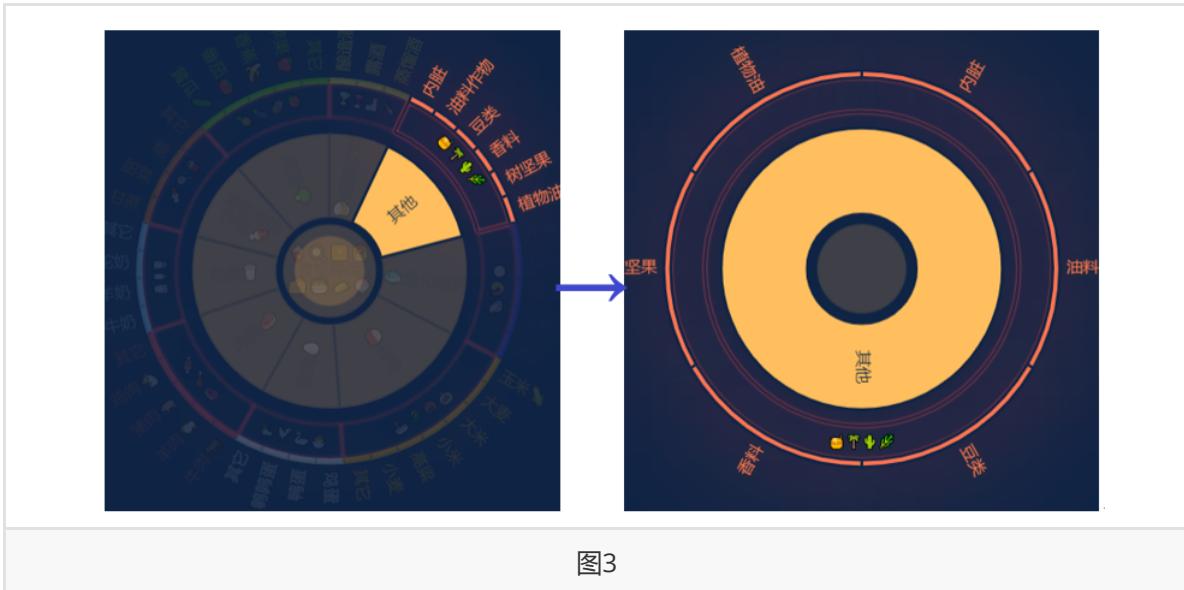


图3

4.3 视图③



图4

视图③为地图-柱状图，描述了在视图①选定的摄入类型下，世界各国在视图②选定的食物来源的占比的地理分布。颜色深浅表示数值的大小，越深色表示占比越大。地图方面允许用户通过鼠标移动到相应的地区产生高亮，同时显示具体国家信息和占比，如图4所示。点击之后变为红色，将选定国家或地区；地图还支持相应的拖拽与缩放。用户可以根据颜色分布的集群来分析地理位置因素引起的占比差异。地图采用echarts编写，在地图方面echarts与d3的功能基本无差异，但echarts在颜色坐标尺拓展方面有更大优势。在左下角的标尺用户可以直接选定数值来显示对应的国家，也可调整颜色范围来观察相应数值范围的国家，如图5所示。



图5

地图右上角的四个按钮依次为：下载、刷新、切换条形图、切换地图。下载按钮可以支持用户下载当前视图③为png图片格式保存；刷新可以支持撤销当前在地图所作的一切更改；切换条形图可以将当前地图变换为柱形图显示；切换地图为将当前柱形图切换为地图。

当前地图的柱形图显示如图6所示，设计柱形图的原因在于渐变色地图对于表示相近数值较为模糊，难以区分大小。使用条形图变换可以让用户查看具体的细致差异，通过滚轮缩放或者点击具体柱形可以进一步查看柱形图的细节，如图7所示。柱形图采用echarts编写，使得在地图与柱形图之间的动画过渡较为炫酷且方便。

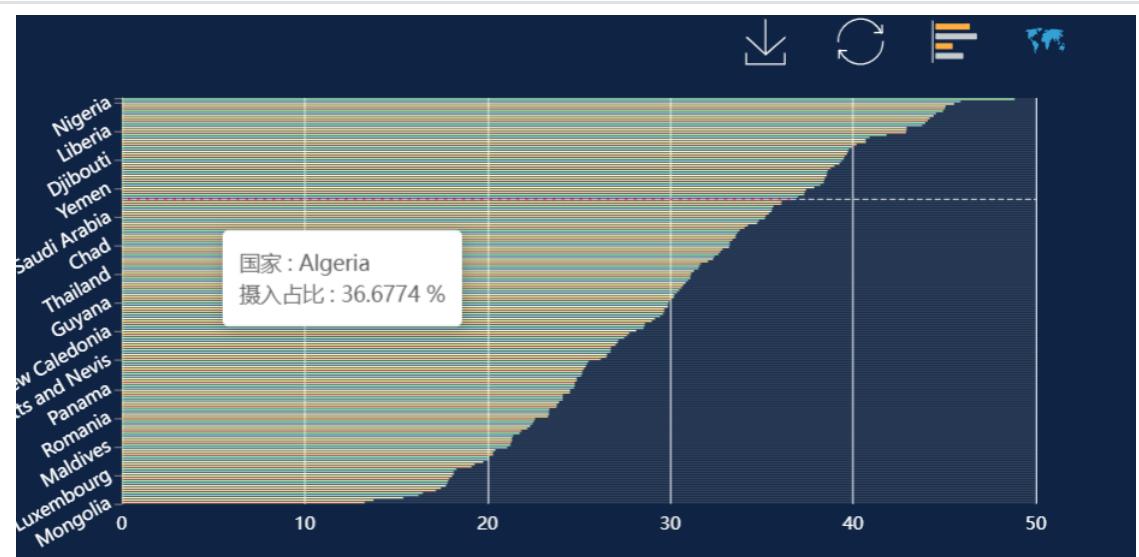


图6

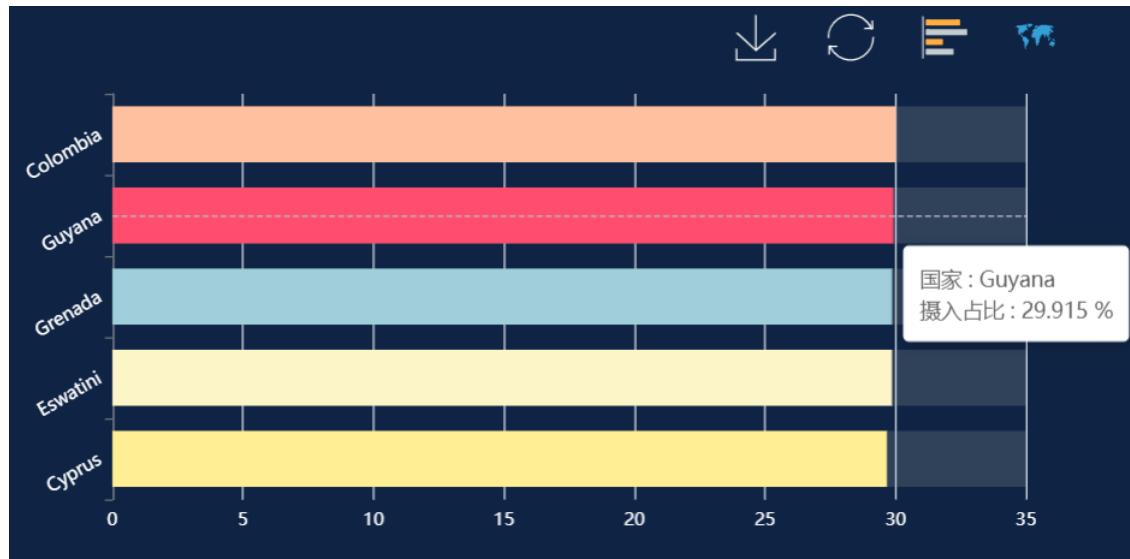


图7

在视图之间的交互方面，地图和条形图所展示出来的具体视图是视图①和视图②共同作用的结果。视图①选定摄入类型，例如“脂肪摄入量”，视图②选定食物种类，例如“蔬果”，那么视图③将展示“世界各国脂肪摄入量来自于蔬果类食物占比”情况，视图①和视图②任意一个变换都可以修改视图③。视图连接采用 JavaScript 的全局变量、显示回调以及视图重画的方式，视图①点击事件会修改读取的后台文件名，视图②的点击事件则修改读取文件的相应的列(即种类)，两者的点击事件都会将 echarts 的相应视图读取的数据来源进行更新，并且重画在相应 div 中。



图8

4.4 视图④

视图④为矩阵树图，描述的在相应国家在某种摄入类型下的食物种类来源情况，如图8所示。用户可以通过视图④查看一个国家或地区的饮食组成结构，将鼠标移动到具体矩形可显示具体信息的标签。由于空间限制，个别占比较小的矩形可能无法展示完全信息，通过滚轮缩放或点击相应矩形查看更加细致介绍，底下显示当前目录，如图9所示。编码采用echarts，echarts在矩阵树图有更多完善的交互功能，包括视图缩放以及动画过渡。

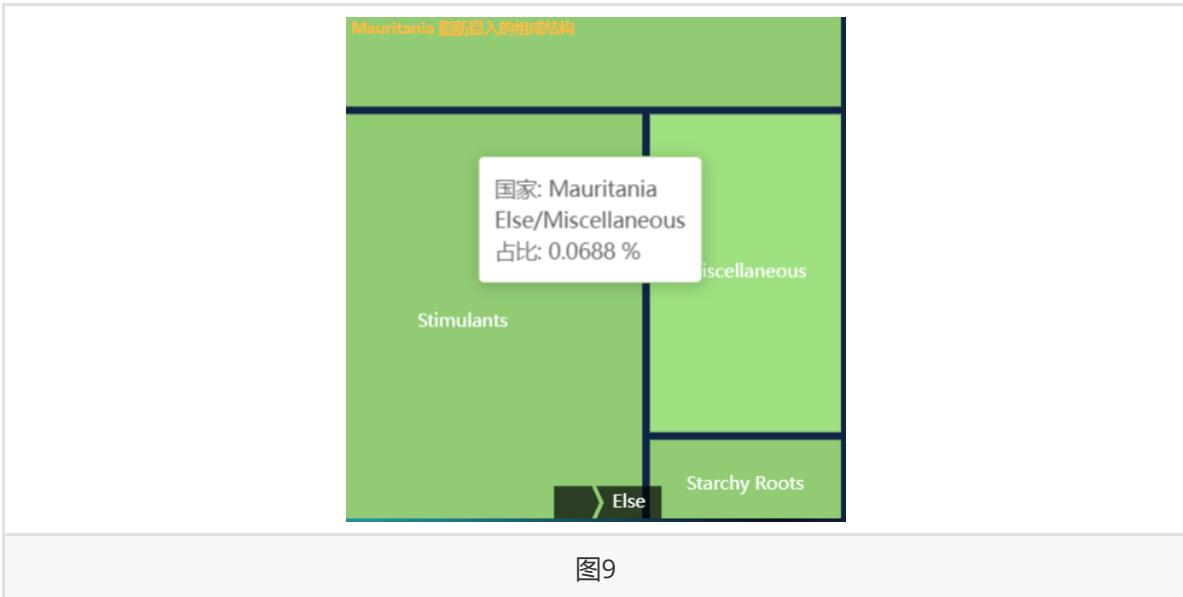


图9

视图④是视图①和③共同作用的结果，视图①选定摄入类型，例如“脂肪摄入”，视图③选定地图或条形图上的国家，如“中国”，那么视图④将展示“中国的脂肪摄入组成结构”。同样采用JavaScript全局变量以及视图重画的方法，视图①点击事件会修改读取的后台文件名，视图③的点击事件会读取相应国家名称传给视图④，视图④在相应文件的对应行找到所有信息完成矩形树图的重画。

4.5 视图⑤



图10

视图⑤是堆叠条形图的纵向排列，描述了世界肥胖率或营养不良率(前十个)的分布以及各个国家的食物组成结构，如图10所示。视图⑤帮助了解世界前十个国家的肥胖率(从高到低)或者营养不良率(从低到高)(通过change order变换)的排列情况，同时使用堆叠条形图帮助用户了解该相应国家的食物来源组成结构。受限于空间大小，视图⑤的具体形状难以伸展。通过视图左上角的“跳转”可以看到更加具体的视图⑤，如图11所示。为了便于用户进行对比，用户点击相应条形图可将各条形图扩展到相同长度，便于进一步比较相应食物组成的占比大小差异，如图12所示，双击相应条形即可恢复初始模样。

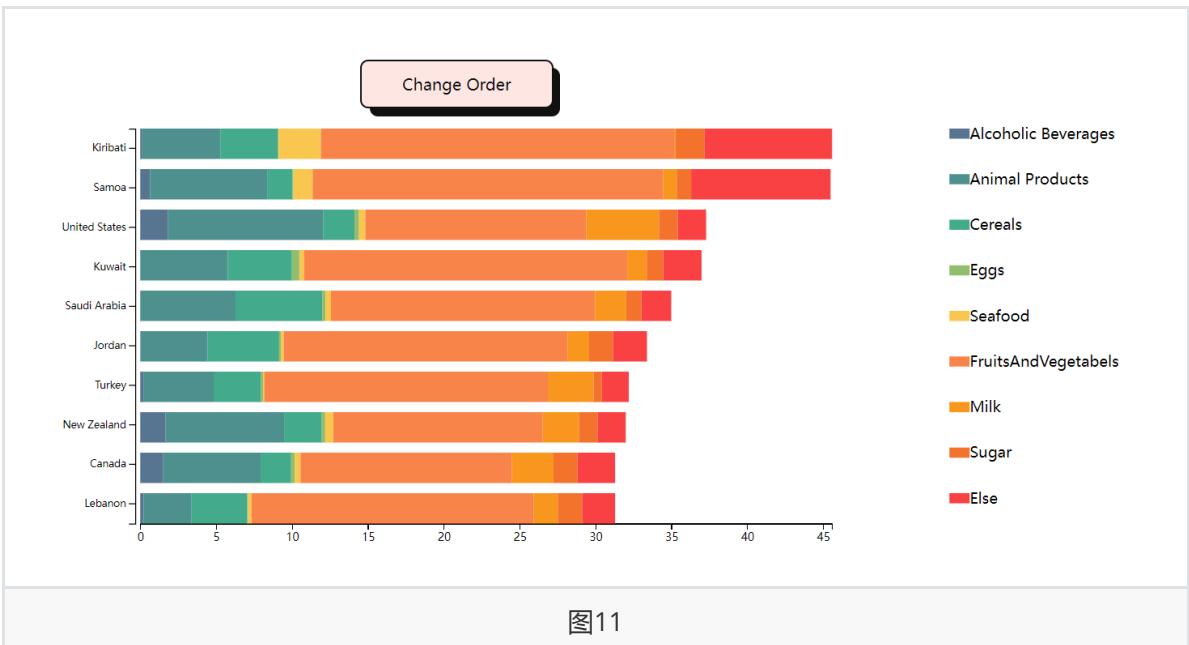


图11

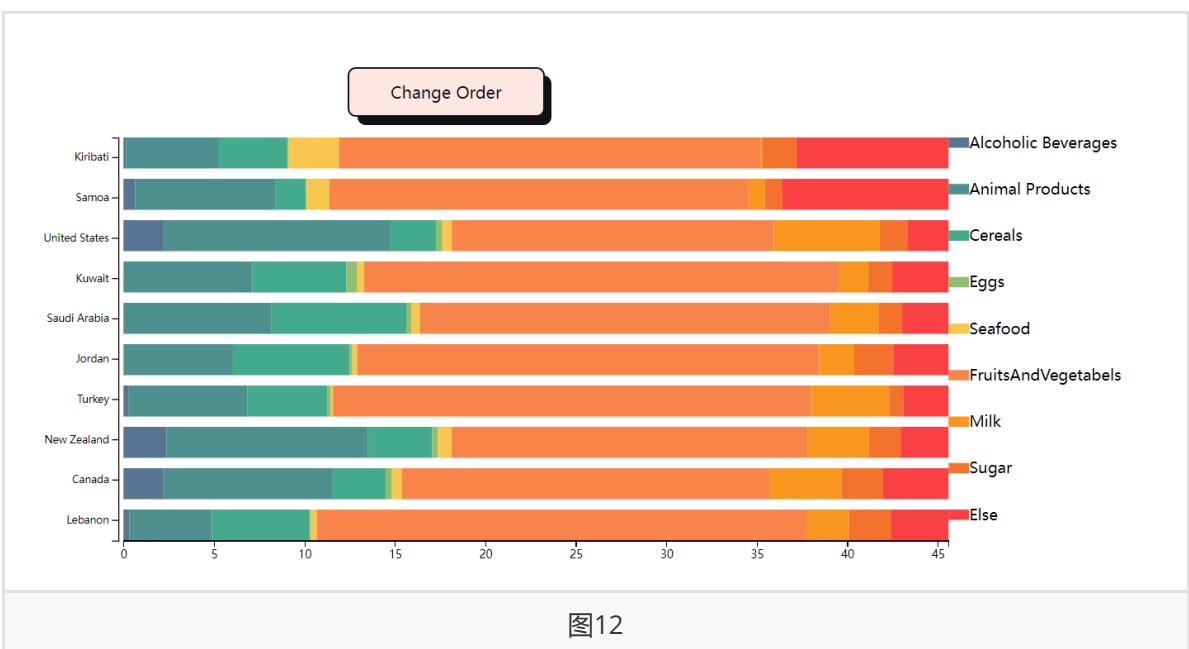


图12

编码实现方面使用d3实现，d3相比echarts偏向于底层，难度更大，但自由度也更高，可以根据实际需要定制所需要的视图功能。对于视图⑤在堆叠条形图基础之上实现相同长度下占比对比这一需求，使用d3通过自定义对应的函数设置具体的矩形长度、伸展位置及动画等参数可以更好地满足实际需要。

4.6 视图⑥和视图⑦



图13

视图⑥和视图⑦为基于源数据使用机器学习得到的肥胖率预测模型，用户可以根据左边的8个仪表盘调整相应的数值来预测患肥胖的概率。用户交互方式为通过鼠标在对应的仪表盘上点击数值，右侧的预测数值及可视化结果就会根据用户输入进行调整变化。

视图①~⑤用于帮助用户分析已有数据的特征及结果，推测及探究相应结论。视图⑥和⑦用于帮助用户基于现有数据预测未来发展趋势，提供启发式思路解决实际问题。

5 反思

5.1 项目发展概括

在第一周时，本组成员都处在寻找数据集和选定主题的阶段。这个阶段，本组成员找了很多数据。思考过这份数据的主题和内容是不是我们想要的，如果利用这份数据，我们能够做出什么成果来，怎么实现，呈现的效果将会是怎么样。我们曾经选出过一份关于轨道卫星的数据，奈何于这份数据数据量过大，且内容繁杂，最终没有选用。曾经也挑选出过一份全球程序员薪资数据，此数据数据量太小，可实现的点并不多，因此也未选择该分数据。在人民生活日益富足的情况下，饮食健康逐渐成为了一个人人关心的话题。因此，我们针对饮食健康这一主题，最终于Kaggle上得到了一份全世界各国饮食习惯的数据。这复合我们的预期主题，且是一个比较有意义的话题。令人开心的是，本数据集数据量适中，不会过于繁杂，也不会让人感觉没有可视化前景。因此，我们选择了本份关于全世界各个国家饮食结构的数据。选定主题后，第一次提交时，每个人仅仅将自己负责的那部分图绘制出来，还没有实现图与图之间的交互，经过不断的试错与调整，最终将所有人绘制的图合并在一个网页上，并且相互之间有着良好的交互效果。

各阶段可视化界面如下。





阶段二可视化界面



阶段三可视化界面

5.2 视觉化目标的变化

针对地图，在对地图可视化时，初始视觉化目标是形成3D地球仪，实现可以自由拖拽旋转地球。各个国家或地区的value在地球仪上的显示方案为半径方向的柱形高度。这样整体的可视化结果类似于“刺猬球”一样，可以同时查看对于数据的集群分布以及地区具体特征。但实现时发现有一些问题：个别国家和地区的柱形选择为圆柱或者立方体时虽然高度的极值差异化明显，但相近值的差异难以分辨；柱形的宽度难以统一设定，个别地区过宽的柱形显示会互相遮挡，过窄则可视化特征不明显，于是采取了目前2D的效果。

针对堆叠图，起初时想要以类似时间线的形式，将想要可视化的数据逐步展开。在鼠标悬浮上每一列时，都会将该列的数据对应展开。后来选择以比较常规的堆叠图，坐标轴的形式呈现。除了基本的数据表述功能外，本堆叠图提供了额外的数据比较功能，可以将不同长度的堆叠条拉伸到同一长度，便于比较不同堆叠条之间比例结构，有利于分析不同饮食结构对肥胖率的影响。

针对矩阵树图，起先是想做两个图轮流切换，这两个图分别是南丁格尔玫瑰图以及矩阵树图，但是读取数据集后进行可视化时发现，属性之间的值普遍差异较大，因为数据集与能量、脂肪摄入有关，不可避免有一些属性的值相较于其他会非常小（例如糖类），而有一些又会显得很大（例如肉类），这些数值近乎为零的属性的存在使得南丁格尔玫瑰图呈现出来的效果不尽人意，在中期报告中提到以雷达图的形式来代替南丁格尔玫瑰图，但是在最终实现时，感觉到这一雷达图的存在显得有些冗余，即不需要两个图轮流切换，因为矩阵树图已经足够表现出相应国家相应摄入类型的饮食结构。

5.3 技术化目标的变化

起初打算学习前端框架React，并且已经复习了基本的JS知识，学习了一些React知识。后来发现，不能系统地学习知识再去使用，应该遇到不会的问题时再探索相关的技术。学前端React的目的是，我们需要跟踪变量，在变量改变时，实现页面的同步刷新。后来还是决定使用基本的页面布局和d3，因为d3可以利用土方法实现类似React的效果，基本的HTML + CSS 布局也能满足需求。若无时间限制，我们可能会呈现一套以React为框架的可视化成果。在实现地图方面，地球仪本质上是贴满纹理的圆形，无相应实用函数，必须渲染纹理以及手动定位国家地区，各个地区分界明显程度取决于纹理图的清晰程度；实时渲染难度大、速度较慢等等。最后选择了平面可拖动、缩放地图，并可通过变换为条形图来保存初始可视化目标。

5.4 就D3技术上的可能性而言，你最初的建议有多现实？

d3技术具有可视化地图、条形图的相应功能，并且在技术基础上实现较为成熟。但对于地图和条形图之间的切换为瞬变，较为生硬；同时在条形图显示上，如果条形图数据量较大会导致信息的不完全，设置缩放可以解决部分问题，但如果设置平滑动画则较为复杂的。针对矩阵树图那部分，d3实现起来很方便，可视化第一次实验已经接触到以矩阵树图去可视化数据，但是当时没有像现在这样以多层级的形式呈现（点击Else可以查看Else包含的所有属性的新矩阵树图），现在是通过Echarts实现这一效果，但是相信通过d3一定也能做到，后续有机会可以转换为d3进行完善。

5.5 针对未能实现的目标所采取的变通

针对堆叠图，如前文所说想以类似时间线的形式，将想要可视化的数据逐步展开。在鼠标悬浮上每一列时，都会将该列的数据对应展开。后来无法想出利用d3的实现方式，选择了更利于进行结构比较的堆叠图。在进行堆叠图的绘制时，我在设计交互效果时，无法将一个堆叠条和一个堆叠条中的矩形块进行区分。无法实现当鼠标悬浮在一个矩形块上方时，显示该矩形块的属性的交互效果。并且，由于有些矩形块比较小，可能也不利于信息的显示。最终，以添加图例的方式解决这一问题，反而觉得视觉上比初始的想法要更好。

5.6 如果重新开始，会采取什么不同的做法？

如果时间充足的话，我们团队应该会首先学习一个适合的前端框架，如React。然后选择一门与d3类似的画图语言，或者就利用d3，做出从用户角度考虑的页面和可视化效果。

6 团队评估

李隆华：创新地将食物种类以旭日图的形式呈现出来，绘制仪表盘并提供鼠标点击改变数值的交互，构建简单的机器学习模型，提供与地图交互的接口。

许凌艺：完成地图、条形图的绘制以及两者之间的动画转换，绘制摄入类型菜单栏，提供与食物种类、矩阵树图交互的接口，并将其与其他队员提供的接口进行对接。

王一凡：完成矩阵树图、雷达图（最终因数据集的呈现效果不好而放弃）的绘制，提供与摄入类型菜单栏、地图交互的接口，对主页的边框形式、背景、配色进行了处理。

倪诗宇：数据预处理，完成堆叠图的绘制，完成了深度学习模型训练（点击仪表盘需要两秒左右才出结果，最终因响应过慢无法提供实时交互而放弃），对主页各区域的排版进行了处理。

7 结果缩略图

见根目录下文件thumbnail.png。

