

# Weekly Report in Week 1 July 2022

---

一周各种零碎的事情:

前往出入境大厅办理逗留签注，等待办证完毕将带签注的港澳通行证寄到南科大。

办理毕业一系列相关手续，清理宿舍办理退宿，目前于7.4下午全部办完（应该没有问题了）。

目前已将所有所需材料（由学校密封的最终成绩单、毕业证和学位证的中英文的电子版，还有e-visa）邮寄或电邮给了港城那边，等待院系秘书来核实是否满足offer 条件。

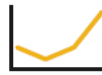
## 1. Book Notes

---

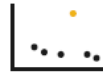
- Visualization Analysis and Design
  - Chapter 3
    - 书接上周，对于动作（Action）包括：分析，搜索和查询
      - 查询包括识别，比较和总结，三者不同在于，目标的范围，单一，多个还是全部目标。不过，随着目标越多，对设计者的编码要求也就越困难，如何在有限空间筛选目标，展示目标是一个问题。对于summarize (overview)，通常是用**一个文本**（动词，名词），可以给用户提供一个全部的视角，并能够总结所展示内容的概括总结。
    - 对于目标（Target），是数据中对用户所感兴趣的所在。

➔ All Data

➔ Trends



➔ Outliers



➔ Features



➔ Attributes

➔ One

➔ Distribution



➔ Extremes

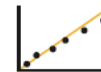


➔ Many

➔ Dependency



➔ Correlation



➔ Similarity



➔ Network Data

➔ Topology



➔ Paths



➔ Spatial Data

➔ Shape



- 对于所有数据：**trend**，趋势是一个数据的模式的高级特征，比较概括的描述数据的整体走向。但是总会有些数据并不在这趋势之中，描述他们只能用outliers（异常值）。特征（feature）针对任意有趣的特殊结构。另，trend同义有pattern，outliers同义有anomalies，novelties，deviants和surprises
- 对于属性来说，单一属性对值有他的分布（distribution）和极值（extremes）来描述。多个属性间也有关系来描述，之间的依赖（dependency），相关性（correlation）和相似性（similarity）：依赖为第一个值直接依赖于第二个；相关性则为多个属性中，每个都与前一个关系紧密相似性则为在某一个可测量值上，他们的全部值都有所相似，当然程度是自己定的。
- 另外对于网络结构数据和空间数据，拓扑链路和几何图形才是面向用户的重点。
- 作者于该章先粗略的给出一个怎样完成前三章个任务的途径（How），从而统摄接下来的各个章节。

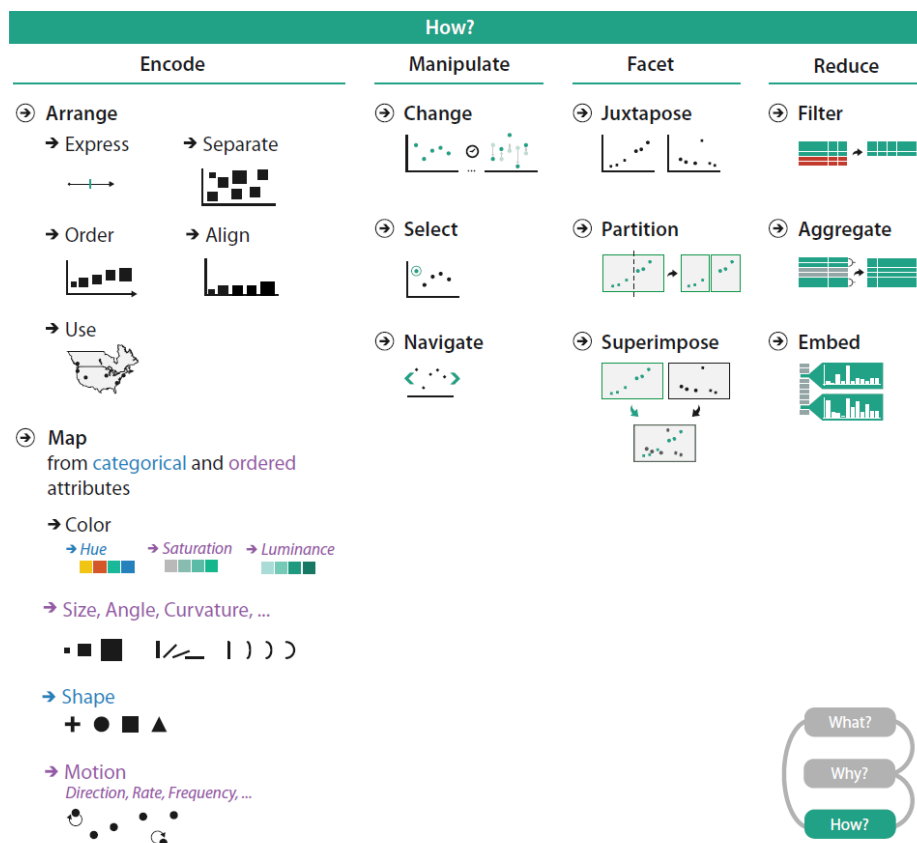
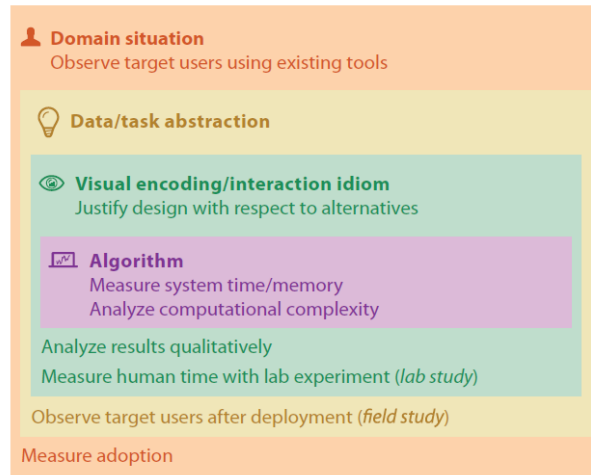


Figure 3.7. How to design vis idioms: encode, manipulate, facet, and reduce.

- 对于如何构建vis的语言，作者在high-level上分了编码（encode），操控（manipulate），排版/组合（facet）和简化（reduce）
  - 对于如何编码又分安排（arrange）和映射（map）
    - 其中安排有5个选择来从空间上管理数据：**表达值**（express），**拆分、排序、对齐区域**（separate, order, align）以及**使用已有的空间数据**（use）
    - 对于没有空间特征的那一部分，可以关注他们的颜色，大小，角度和形状，当然还有动作
  - 对于操控：涵盖了视图各方面的改变（change），选择存在于视图中的元素（select）和给变化来导航
- 三个例子来展示如何正确是由what-why-how三步
  - 第一个是比对分析：对于目的和数据一致的，如何编码呈现就出现不同
  - 第二个是派生单一属性：数据过于繁多，可以通过整合派生的方法压缩数据的同时，更好的表达数据，更进一步，可以将派生数据加入输入，将目的变更为总结和拓扑。
  - 第三个是派生多个数据，如同2中运用派生数据，明确动作和目标（总的来说是task），用多种手段来实现视觉编码。对于任务明确动作和目标，从4个高级途径来找到如何实现这一vis的方法。

#### ○ Chapter 4

- 对于设计vis，作者给出了一个四层递进的分析验证方式：



**Figure 4.1.** The four nested levels of vis design have different threats to validity at each level, and validation approaches should be chosen accordingly.

这一过程先确定应用的特定情况 --> 确定所用数据和构建vis的任务所在 (what, why) ---> 如何进行视觉编码和交互从而更好表达 ---> 设计算法, 尽可能用纯计算形式解决问题。

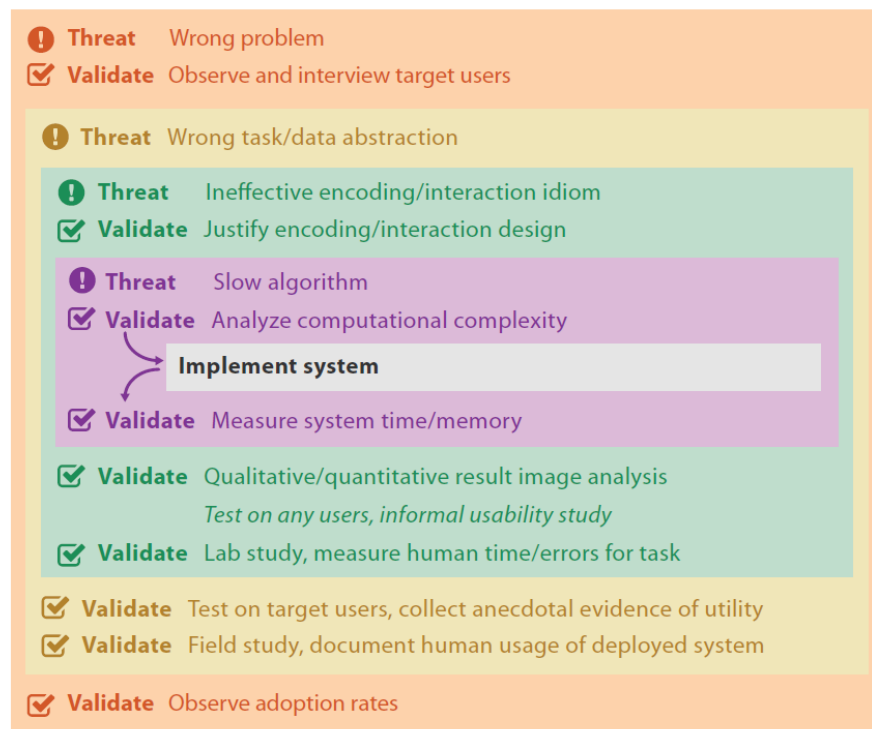
这样的好处在于: 可以独立于构建vis过程中决策的顺序, 分别分析每一层是否正确达成目标, 以防上端出现问题, 下端白费功夫。但无论如何, vis的构建是艰难的, 要不断的决策, 重新决策。。

- 对于第一层这一块 (block) 特定领域的情况 (domain situation)。用户感兴趣的特定领域称为domain。对于识别domain情境, 一般可以通过采访, 观察, 调查来了解用户所需。(通过不断对特定目标接近, 逐步提炼的方法又称用户为中心的设计 (user/human-centered design) )。

**切忌通过假设来瞎设计, 需要真正的接近用户。** 总之要足够具体的一个目标问题才可以导向下一层级

- 对于第二层这一块, 也就是chapter 1 2 3里面说到的what和why --> 数据和任务抽象化。这一设计需要将上一层中非常具体的某一领域的问题, 通用化的表达出来 (即使相差甚远情形的问题也可以抽象到同一vis工具中。)
  - 任务抽象, 设计者通过标识
  - 数据抽象, 需要将源数据转换, 来方便表达进而解决用户的问题。即使一些数据集已经可以直接满足视觉编码来解决问题, 但是总归转换形式是更好的。
  - 另外, 数据和任务的抽象一定要有明确的选择和理由, 否则没有正当理由的所谓抽象化, 只会让用户感到更加莫名其妙, 对于vis更加混乱。
- 对于第三层这一块, 涉及到视觉编码和交互语言。到了这一步, 设计者需要从特定方法来创造, 控制一个受之前任务导向, 由之前所选抽象化数据的视觉编码。**作者在此将实现这一过程的各种方式称为idiom。**关于idiom design 有两个: 将数据用图像来展示, 掌控了用户所见 (**visual encoding**) ; 动态操作所要表达的东西, 其负责用户可以自行改变所见 (**interaction**)。你做出好与坏的设计决定取决于是否理解人类认知与记忆 (既是认知科学, 也是一种经验)
- 对于最后一块算法, 通过数学模型计算, 自动解决上述问题。但切记它不是用来识别上面这些块的, 而是设计他们的 (所以识别和设计区别何在?)。
- 总之这样的嵌套模型讲究的就是将算法和语义设计分离设计。
- 作者又从攻关的角度上对vis进行分类, 分为**问题驱动** (从顶向下) 和**技术驱动** (从底到上)。总的来说, 这本书的框架的重点还是在抽象化和视觉编码, 交互idiom上面。这样就可以在学习过后, 有能力灵活选择角度来完成vis。

- 对于问题驱动，就是从一个特定领域问题出发，分别逐级对抽象化，idiom和算法做出抉择。用户遇到问题，有了需求就解决。这种模式大多使用现有的idiom和算法来力求高效解决问题。（design study），所以大部分挑战都是在抽象化部分，
- 对于技术驱动，首先要致力于idiom和算法的工作，即当你自己创造这些，将会更好的支持上一块内容的完成。
- 这一部分开始将从总到分的讨论如何验证vis的有效性。基于前面的嵌套结构，将分块来依次验证他们。对于不同层级，威胁vis的原因也不一：
  - 对于错误的问题：可能是你误解了人家的需求！
  - 对于错的抽象化：你在给用户看错的东西！
  - 对于错的idiom：你展示的方式有问题！
  - 对于错误的算法：你的代码太慢辣！
- 至于验证的途径，粗略的步骤可以参考文中所给视图：



**Figure 4.5.** Threats and validation at each of the four levels. Many threats at the outer levels require downstream validation, which cannot be carried out until the inner levels within them are addressed, as shown by the red lines. Any single project would only address a subset of these levels, not all of them at once.

根据图和内容，区分出了**直接验证**和**下游验证**两种途径。

- 层级越高的部分，验证的方式大多都不是立即验证，因为需要得到下游层级的结果。但对于下游验证来说，很有可能由于本层级的问题，反倒吸引他人关注了更上端层级的验证，增大了迷惑和难度。
  - 对于最高层级的验证：domain validation，最基本的问题通常源于这个问题被错误的特征化了，说直白点就是压根没搞懂人家用户想要啥，就说vis可以辅助解决这个问题，其实人家问题都压根不在这。
    - 如果是直接验证的话，就立马去观察或者采访目标用户。--> 对于这种方式通常的实践途径是特定实地学习（field study），亲自观察调研。--> 而对于field study，又以访谈为优，单独的静默观察无法适应复杂的认知任务，眼见也不一定的为实，可以先观察，再及时打断并提问。
    - 如果是下游验证的话，汇报vis被目标受众所采用的速度。尽管这一指标不能完全反应vis的全部，但是它也可以作为一个信号，来汇报用户所自愿做的事情，而非是要求他使用。

- 对于抽象化层级的验证：abstraction validation，其会出现的问题在于这些抽象化了的任务和数据，没法解决上一层的具体问题。解决其关键在于系统必须通过**目标用户的考验**（可以完成他们的一工作），**而非是vis设计系统可以特定完成的任务**。
  - 第一个比较普遍的下流验证途径是找一群目标用户来尝试vis，收集一些证据来表示这玩意确实有用。这些“轶事”可以使观察发现，假设验证啥的。（这种方法必须在其他层级全部解决之后才能实现）。作者再次说到，**一定要区分用户发现和设计者发现，前者才是关键！**
  - 另一个更严谨的途径是进行上面提到的field study --> 观察并记录目标受众如何使用已部署的系统。这一层的field study 与之前不同的是，**用户介入到了系统的部署当中**，而不是直接观察他们实际使用之类的。
- 对于idiom validation，这一层会出现的问题主要在于所选择的这些表达没法向用户传达他们渴求的抽象。
  - 第一种直接验证方式为，仔细的，基于已知的感知和认知准则来辨别、评判idiom的设计。
  - 对于下游验证，一种是 lab study，在实验室中进行可控实验。（**值得一提，user study分为 lab 和field study，目前两种区别在于观察实验的场所不同**）。这一方法适合通过检测人们对于设计者所选给用户的抽象任务中的表现来梳理选用特定idiom。这样的实验中就会有定性和定量检测。比如非常常见的参与者所耗时间和产生的错误。
    - 定性主观上，用户偏好等。定性数据的收集通常包括要求参与者通过**问卷**来表达他们的策略。在这种情况下，人类行为的预期变化足够小（被限制在问卷里了）
    - 定量测量上，定量数据，比如vis工具记录鼠标移动和点击等操作，或者使用外部设备跟踪参与者的眼球运动。
  - 另一个下游验证方式，以静止图像或视频的形式对结果进行展示和定性讨论。当有明确的讨论并指出所求的特性时，这种方式是最好的，不用作无辅助的推断。作者反复说，假设、臆测啥的尽量避免。
  - 第三个下游方式为，对实现的系统中的生成的结果图来定量检测。又名 quality metrics
  - 另外作者也提到，非正式的可用性研究很重要，但不成为验证的一个方法。很重要且被列在这一层是因为：如果系统不可用，就无法从用户研究中得出有关其效用的有用结论
- 对于算法验证这一层：algorithm validation，其最主要的问题出自，该算法在时间或内存性能方面是次优的，要么是不如理论最小值，要么是不如以前提出的算法。尤其是时间慢，直接影响交互的响应时间
  - 一个直接验证的方式为，分析算法的复杂度。可以从数据集来分析复杂性，也可以直接从显示的像素数
  - 下游验证的方式为，测量系统的挂钟时间（进程从开始到结束时间？突然操作系统。。）和内存性能。数据集大小层面如何影响算法，直接上benchmark来评估。
  - 另一个**威胁**是算法级别的不正确，即算法的实现不符合idiom级别以上的规范。问题可能来自糟糕的算法设计，或者算法的实现可能像任何计算机程序一样有bug。

算法不正确的威胁往往是隐式解决而不是明确的。呈现由实现的算法创建的静态图像或视频是针对这种威胁的隐式验证的一种形式，在这种验证中，论文的读者可以直接看到算法正确性目标已经满足。为什么这些图像表明算法实际上是正确的明确性讨论并不常见。

## 2. Papers

---

Null

## 3. Some New Words

---

- hue: a color or shade
- unfettered: unrestrained or uninhibited.
- corollary: a proposition that follows from (and is often appended to) one already proved.
- conjecture: an opinion or conclusion formed on the basis of incomplete information.

## 4. Course of Zhejiang University

---

听完了这周全部的理论部分和前沿讲座

顺便阅读（白嫖）了宽知善视上的课堂总结笔记。。。