These are the links in this lecture:

* + [Enterprise Data Analysis and Visualization: An Interview Study](http://idl.cs.washington.edu/files/2012-EnterpriseAnalysisInterviews-VAST.pdf)
  + [Data Scientist: The Sexiest Job of the 21st Century](https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century/ar/1)
  + Ben Fry's [keynote talk for VIZBI 2010](http://vizbi.org/Videos/11551146)
  + [EMBO](http://www.embo.org/) workshop on visualizing biological data (http://vizbi.org)
  + [Ben Fry](http://benfry.com/)
  + [Ben's keynote](https://youtu.be/bGseXNCR1j0)
  + 2009 Above the Clouds [Technical Report](https://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf)

CONTRAST: DATABASES

This comparison may seem like we are just comparing two different types of databases, relational (strongly structured) databases and NoSQL (semi-structured) databases, instead of a comparison of databases in general versus data science. Indeed, relational databases can contain terabytes of data and many people perform data science on data retrieved from relational databases.  However, as we will see in Week 3's lectures, much of the information that is typically used for data science is unstructured or semi-structured, and cannot be stored in a relational database.

The comparison is complicated because the strongly structured nature of relational databases means that they are often used to store supplementary information to the data available in NoSQL stores. For example, an ecommerce website would typically store credit card information in a relational database, while it could store product recommendation information in a NoSQL store. Thus, in this environment data science would involve data retrieved from both strongly-structured and semi-structured stores.

In this [keynote talk for VIZBI 2010](http://vizbi.org/Videos/11551146), the [EMBO](http://www.embo.org/) workshop on visualizing biological data (http://vizbi.org), [Ben Fry](http://benfry.com/) discusses principles of graphics design and of dynamic visualization that can improve the insight gained from data. He also presents a range of visualizations of genetics data he has created, some of which have been used to improve tools used by biologists, some of which has been exhibited in art galleries and in Hollywood movies. He also presents 'Processing', a computer language he co-developed that can easily create dynamic visualization of complex data. A shorter version of Ben's keynote can be viewed on [here](https://youtu.be/bGseXNCR1j0) on YouTube.

# Above the Clouds: A Berkeley View of Cloud Computing

This 2009 [Technical Report](https://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf) defines Cloud Computing terms, presents an economic model that quantifies the key buy vs. pay-as-you-go decision, offers a spectrum to classify Cloud Computing providers, and gives our view of the top 10 obstacles and opportunities to the growth of Cloud Computing.

The analyst archetypes in this video module are just one example of how we can categorize different analyst roles. You can find another set of three analyst archetypes in the optional reading, [Enterprise Data Analysis and Visualization: An Interview Study](http://idl.cs.washington.edu/files/2012-EnterpriseAnalysisInterviews-VAST.pdf), where they define the roles of hackers, scripters, and application users:

* + **Hackers** are the most proficient programmers of the three groups and the most comfortable manipulating data. They typically used at least three different types of programming languages. In addition to working with an analysis package (e.g., R or Matlab), they frequently used a scripting language (Python, Perl) and a data processing language (SQL, Pig, etc).
  + **Scripters** perform most of their analysis within a software package such as R or Matlab. They were able to perform simple manipulations such as filtering and aggregating data, but typically could not perform custom operations such as parsing log files or scraping data off the web. They generally operated on data that had been pulled from the data warehouse by IT staff and stored in an expected format. Some of these analysts could write simple SQL queries (e.g., without joins) to pull data into their analytic tool of choice. In some cases, they were comfortable writing scripts in a scripting language, but typically do not know how to create scripts that run at scale.
  + **Application Users** perform almost all operations in a spreadsheet or other dedicated analysis application (e.g., SAS/JMP, SPSS, etc). Like scripters, they typically required someone to prepare data for them by pulling it from the warehouse.