DSC 201: Data Analysis & Visualization

Data & Tables

Dr. David Koop
http://www.cis.umassd.edu/~dkoop/dsc201
Course Material

- Course Website
  - All material will be posted there
  - myCourses for turning in assignments

  - Good reference for data science topics in Python
  - McKinney created the Pandas package
Course Material

• Textbook: *Python for Data Analysis* by Wes McKinney, 2nd ed., 2017
  - 2nd ed. not available yet
  - Prerelease chapters available through myCourses
  - Password
  - Please do not share

• Other references:
  - *Python Data Science Handbook*, J. VanderPlas
  - learnpython.org
Course Material

• Software:
  - Anaconda Python Distribution (https://www.continuum.io/downloads): makes installing python and python packages easier
  - Jupyter Notebook: Web-based interface for interactively writing and executing Python code
Course Material

- Pandas:
  - Python library for data analysis
  - Many operations available
  - Efficient

- Tableau:
  - Desktop (or web) application
  - Create visualizations quickly

- Other Visualization Tools:
  - Python libraries: Matplotlib, Bokeh, folium
  - Don't have to move between applications
Grading

- Assignments (5): 45%
- Tests: 2 in-class: 15% each, 1 final: 20%
- Class Participation: 5%
- Late Policy
Tests

• Test 1: October 5 in class
• Test 2: November 14 in class
• Final Exam: December 13, 11:30am-2:30pm
• Tests may not be rescheduled. Tests can only be made up in case of a documented emergency.
Accommodation Policy

• Please contact me at the beginning of the semester and provide the appropriate paperwork from the Center for Access and Success.
• Please update me if anything changes during the semester.
• Center for Access and Success: Pine Dale Hall Room 7136, x8711, access_success@umassd.edu
Academic Honesty

• Do not cheat!
• You will receive a zero for any assignment/exam/etc. where cheating has occurred. Repeat offenders will fail the course.
• You may discuss problems and approaches with other students
• You may not copy or transcribe code from another source
Course Registration Reminder

• Add/drop deadline is **today**
• Make sure you are signed up for the course
Chicago Food Inspections Exploration

- Based on David Beazley's PyData Chicago talk
- YouTube video: https://www.youtube.com/watch?v=j6VSAsK Aj98
- Our in-class exploration:
  - Don't focus on the syntax
  - Focus on:
    - What is information is available
    - **Questions** are interesting about this dataset
    - How to decide on good follow-up questions
    - What the computations mean
Data

- What is data?
  - Types
  - Semantics
- How is data structured?
  - Tables (Data Frames)
  - Databases
  - Data Cubes
- What formats is data stored in?
- Raw versus derived data
Data

• What is this data?

<table>
<thead>
<tr>
<th></th>
<th>Address</th>
<th>1st Column</th>
<th>2nd Column</th>
<th>3rd Column</th>
<th>4th Column</th>
<th>5th Column</th>
<th>6th Column</th>
<th>7th Column</th>
<th>8th Column</th>
<th>9th Column</th>
<th>10th Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>R011</td>
<td>42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>00001455</td>
<td>00000134</td>
<td>00033341</td>
<td>00071255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R170</td>
<td>14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>0011051</td>
<td>00000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R046</td>
<td>42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>0007908</td>
<td>00000323</td>
<td>0001183</td>
<td>0003001</td>
<td>00040759</td>
<td>00096613</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Semantics: real-world meaning of the data
• Type: structural or mathematical interpretation
• Both often require metadata
  - Sometimes we can infer some of this information
  - Line between data and metadata isn’t always clear
### Data

<table>
<thead>
<tr>
<th>REMOTE</th>
<th>STATION</th>
<th>FF</th>
<th>SEN/DIS</th>
<th>7-D AFAS UNL</th>
<th>D AFAS/RMF</th>
<th>JOINT RR TKT</th>
<th>7-D UNL</th>
<th>30-D UNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R011 42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>0001455</td>
<td>00000134</td>
<td>00033341</td>
<td>00071255</td>
</tr>
<tr>
<td>2</td>
<td>R170 14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>00011051</td>
<td>00000827</td>
<td>0003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
</tr>
<tr>
<td>3</td>
<td>R046 42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>00007908</td>
<td>00000323</td>
<td>0001183</td>
<td>0003001</td>
<td>00040759</td>
<td>00096613</td>
</tr>
<tr>
<td>4</td>
<td>R012 34TH STREET &amp; 8TH AVENUE</td>
<td>00188311</td>
<td>00006490</td>
<td>00000498</td>
<td>0001279</td>
<td>0003622</td>
<td>00035527</td>
<td>00067483</td>
</tr>
<tr>
<td>5</td>
<td>R293 34TH STREET – PENN STATION</td>
<td>00168768</td>
<td>00006155</td>
<td>00000523</td>
<td>0001065</td>
<td>0005031</td>
<td>00030645</td>
<td>00054376</td>
</tr>
<tr>
<td>6</td>
<td>R033 42ND STREET/TIMES SQUARE</td>
<td>00159382</td>
<td>00005945</td>
<td>00000378</td>
<td>0001205</td>
<td>0000690</td>
<td>00058931</td>
<td>00078644</td>
</tr>
<tr>
<td>7</td>
<td>R022 34TH STREET &amp; 6TH AVENUE</td>
<td>00156008</td>
<td>00006276</td>
<td>00000487</td>
<td>0001543</td>
<td>0000712</td>
<td>00058910</td>
<td>00110466</td>
</tr>
<tr>
<td>8</td>
<td>R084 59TH STREET/COLUMBUS CIRCLE</td>
<td>00155262</td>
<td>00009484</td>
<td>00000589</td>
<td>0002071</td>
<td>0000542</td>
<td>00053397</td>
<td>00113966</td>
</tr>
<tr>
<td>9</td>
<td>R020 47-50 STREETS/ROCKEFELLER</td>
<td>00143500</td>
<td>00006402</td>
<td>00000384</td>
<td>0001159</td>
<td>0000723</td>
<td>00037978</td>
<td>00090745</td>
</tr>
<tr>
<td>10</td>
<td>R179 86TH STREET-LEXINGTON AVE</td>
<td>00142169</td>
<td>00010367</td>
<td>00000470</td>
<td>0001839</td>
<td>0000271</td>
<td>00050328</td>
<td>00125250</td>
</tr>
<tr>
<td>11</td>
<td>R023 34TH STREET &amp; 6TH AVENUE</td>
<td>00134052</td>
<td>00005005</td>
<td>00000348</td>
<td>0001112</td>
<td>0000649</td>
<td>00031531</td>
<td>00075040</td>
</tr>
<tr>
<td>12</td>
<td>R029 PARK PLACE</td>
<td>00121614</td>
<td>00004311</td>
<td>00000287</td>
<td>0000931</td>
<td>0000792</td>
<td>00025404</td>
<td>00065362</td>
</tr>
<tr>
<td>13</td>
<td>R047 42ND STREET &amp; GRAND CENTRAL</td>
<td>00100742</td>
<td>00004273</td>
<td>00000185</td>
<td>0000704</td>
<td>0001241</td>
<td>00022808</td>
<td>00068216</td>
</tr>
<tr>
<td>14</td>
<td>R031 34TH STREET &amp; 7TH AVENUE</td>
<td>00095076</td>
<td>00003990</td>
<td>00000232</td>
<td>0000727</td>
<td>0001459</td>
<td>00024284</td>
<td>00038671</td>
</tr>
<tr>
<td>15</td>
<td>R017 LEXINGTON AVENUE</td>
<td>00094655</td>
<td>00004688</td>
<td>00000190</td>
<td>0000833</td>
<td>0000754</td>
<td>00020018</td>
<td>00055066</td>
</tr>
<tr>
<td>16</td>
<td>R175 8TH AVENUE-14TH STREET</td>
<td>00094313</td>
<td>00003907</td>
<td>00000286</td>
<td>0001144</td>
<td>0000256</td>
<td>00038272</td>
<td>00074661</td>
</tr>
<tr>
<td>17</td>
<td>R057 BARCLAYS CENTER</td>
<td>00093804</td>
<td>00004204</td>
<td>00000454</td>
<td>0001386</td>
<td>0001491</td>
<td>00039113</td>
<td>00068119</td>
</tr>
<tr>
<td>18</td>
<td>R138 WEST 4TH ST-WASHINGTON SQ</td>
<td>00093562</td>
<td>00004677</td>
<td>00000251</td>
<td>0000965</td>
<td>0000127</td>
<td>00031628</td>
<td>00074458</td>
</tr>
</tbody>
</table>
Dataset Types

- **Tables**
  - Attributes (columns)
  - Items (rows)
  - Cell containing value

- **Networks**
  - Link
  - Node (item)

- **Fields (Continuous)**
  - Grid of positions
  - Attributes (columns)
  - Value in cell

- **Geometry (Spatial)**
  - Position

- **Multidimensional Table**
  - Key 1
  - Key 2
  - Attributes
  - Value in cell

- **Trees**

[Munzner (ill. Maguire), 2014]
Data Terminology

• Items
  - An item is an individual discrete entity
  - e.g., a row in a table
• Attributes
  - An attribute is some specific property that can be measured, observed, or logged
  - a.k.a. variable, (data) dimension
  - e.g., a column in a table
## Tables

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Order ID</td>
<td>Date</td>
<td>Order Priority</td>
<td>Container</td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.99</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.69</td>
<td>7/18/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.69</td>
<td>7/18/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.6</td>
<td>6/6/05</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Wrap Bag</td>
<td>0.82</td>
<td>12/23/06</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>5/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
</tr>
</tbody>
</table>
### Tables

**Flat**

- Data organized by rows & columns
  - row ~ item (usually)
  - column ~ attribute
  - label ~ attribute name
- Key: identifies each item (row)
  - Usually **unique**
  - Allows **join** of data from 2+ tables
- Compound key: key split among multiple columns, e.g. (state, year) for population

**Multidimensional**

- Split compound key
- e.g. a data cube with (state, year)

---

[D. Koop, DSC 201, Fall 2017](#)

[Munzner (ill. Maguire), 2014](#)
Attribute Types

- **Attribute Types**
  - Categorical
  - Ordered
  - Ordinal
  - Quantitative

- **Ordering Direction**
  - Sequential
  - Diverging
  - Cyclic

[Munzner (ill. Maguire), 2014]
Categorial, Ordinal, and Quantitative

<table>
<thead>
<tr>
<th></th>
<th>10/14/06</th>
<th>3-Low</th>
<th>Large Box</th>
<th>0.8</th>
<th>10/21/06</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Spec</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.72</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.6</td>
<td>7/18/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.65</td>
<td>7/18/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Spec</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Spec</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Spec</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Spec</td>
<td></td>
<td>0.6</td>
<td>6/6/05</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td></td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td></td>
<td>0.82</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td></td>
<td>0.55</td>
<td>4/19/05</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td></td>
<td>0.38</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td></td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Spec</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Spec</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>4-Not Spec</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
</tbody>
</table>

quantitative
ordinal
categorical
Categorial, Ordinal, and Quantitative

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Description</th>
<th>Quantitative</th>
<th>Ordinal</th>
<th>Categorical</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10/14/08</td>
<td>5-Low</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>0.79</td>
<td>7/17/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>0.65</td>
<td>7/18/07</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>0.72</td>
<td>7/17/07</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>0.58</td>
<td>10/25/07</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>3/19/07</td>
<td>1-Urgent</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>0.23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Specified</td>
<td>0.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Specified</td>
<td>0.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>0.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>4/19/05</td>
<td>2-High</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>1/30/06</td>
<td>3-Medium</td>
<td>0.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>11/28/08</td>
<td>5-Low</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/9/08</td>
<td>2-High</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/10/08</td>
<td>2-High</td>
<td>0.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/11/08</td>
<td>2-High</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/12/06</td>
<td>3-Medium</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/14/06</td>
<td>3-Medium</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>5/3/08</td>
<td>4-Not Specified</td>
<td>0.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>0.64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>9/14/07</td>
<td>2-High</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Attribute Types

• May be further specified for computational storage/processing
  - Categorical: string, boolean, blood type
  - Ordered: enumeration, t-shirt size
  - Quantitative: integer, float, fixed decimal, datetime

• Sometimes, types can be inferred from the data
  - e.g. numbers and none have decimal points → integer
  - could be incorrect (data doesn't have floats, but could be)
Attribute Types

- **Categorical**
- **Ordered**
  - **Ordinal**
  - **Quantitative**

Ordering Direction

- **Sequential**
- **Diverging**
- **Cyclic**

[Munzner (ill. Maguire), 2014]
Sequential and Diverging Data

• Sequential: homogenous range from a minimum to a maximum
  - Examples: Land elevations, ocean depths

• Diverging: can be deconstructed into two sequences pointing in opposite directions
  - Has a **zero point** (not necessary 0)
  - Example: Map of both land elevation and ocean depth

[Rogowitz & Treinish, 1998]
Cyclic Data

3.1. Mathematical description and types of spirals

A spiral is easy to describe and understand in polar coordinates, i.e. in the form \( r = f(\phi) \). The distinctive feature of a spiral is that \( f \) is a monotone function. In this work we assume a spiral is described by

Several simple functions \( f \) lead to well-known types of spirals:

• Archimedes' spiral has the form \( r = a \phi \). It has the special property that a ray emanating from the origin crosses two consecutive arcs of the spiral in a constant distance.

• The Hyperbolic spiral has the form \( r = \frac{a}{\phi} \). It is the inverse of Archimedes' spiral with respect to the origin.

• More generally, spirals of the form \( r = a + b \phi \) are called Archimedean spirals.

• The logarithmic spiral has the form \( r = e^{k \phi} \). It has the special property that all arcs cut a ray emanating from the origin under the same angle.

For the visualization of time-dependent data Archimedes' spiral seems to be the most appropriate. In most applications data from different periods are equally important. This should be reflected visually in that the distance to other periods is always the same.

3.2. Mapping data to the spiral

In general, markers, bars, and line elements can be used to visualize time-series data similar to standard point, bar, and line graphs on Spiral Graphs. For instance, quantitative, discrete data can be presented as bars on the spiral or by marks with a corresponding distance to the spiral. However, since the \( x \) and \( y \) coordinate are needed to achieve the general form of the spiral their use is limited for the display of data values. One might consider to map data values to small absolute changes in the radius, i.e. \( \frac{r}{\phi} \). Yet, we have found this way of visualizing to be ineffective. We conclude that the general shape of the spiral should be untouched and other attributes should be used, such as

• colour,
• texture, including line styles and patterns,

[Sunlight intensity, Weber et al., 2001]
Semantics

- The meaning of the data
- Example: 94023, 90210, 52790, 02747
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
Semantics

• The meaning of the data

• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
  - Salaries?
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
  - Salaries?
  - Zip codes?
• Cannot always infer based on what the data looks like
• Often require semantics to better understand data
• Column names help with semantics
• May also include rules about data: a zip code is part of an address that uniquely identifies a residence
• Useful for asking good questions about the data
Data Model vs. Conceptual Model

• Data Model: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

• Conceptual Model: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    • Quantitative: [32.50, 54.00, -17.30]

[via A. Lex, 2015]
Data Model vs. Conceptual Model

- **Data Model**: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

- **Conceptual Model**: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    - Quantitative: [32.50, 54.00, -17.30]
    - Ordered: [warm, hot, cold]
Data Model vs. Conceptual Model

• Data Model: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

• Conceptual Model: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    • Quantitative: [32.50, 54.00, -17.30]
    • Ordered: [warm, hot, cold]
    • Categorical: [not burned, burned, not burned]
Derived Data

- Often, data in its original form isn't as useful as we would like.
- Examples: Data about a basketball team's games.

  - Example 1: `1stHalfPoints, 2ndHalfPoints`
    - More useful to know total number of points.
    - `Points = 1stHalfPoints + 2ndHalfPoints`

  - Example 2: `Points, OpponentPoints`
    - Want to have a column indicating win/loss.
    - `Win = True if (Points > OpponentPoints) else False`

  - Example 3: `Points`
    - Want to have a column indicating how that point total ranks.
    - `Rank = index in sorted list of all Point values`
Table Operations

- Insert/Delete/Update the data
- Project: get only some of the attributes (columns)
- Select: get only some of the items (rows) via boolean expressions (e.g. attendance > 10,000)
- Join: join two tables together via attributes that represent the same thing
- Project, select, join are key operations in relational algebra
- Group by & aggregation:
  - group all items that share an attribute value
  - compute sum, mean, median, max, min of each group
Databases

• Usually more than one table: keys and foreign keys

• **Database Management System:** software to work with databases

• Optimized storage, security, schema design, administration

• Lots of engineering to process transactions **reliably**
  - ACID (atomicity, consistency, isolation, durability)

• Create new tables, insert new rows, update/delete old rows, query tables for results

• Often coupled with the Structured Query Language (SQL)
  - `SELECT population FROM state_pop
    WHERE state = 'MA' and year = '2016';`

• Other database technologies: in-memory stores, column-based stores, NoSQL, graph databases, XML databases
Data Cubes

- More **dimensions**: not only rows/cols
- aka OLAP cube (online analytical processing)
- Often a **hierarchy** for each dimension
- Hierarchy coupled with **aggregation**
- Example: sales data with a time dimension
  - Examine years, months, or days
  - Find the year with maximum sales
  - Find the average sales per month
  - Find the day with the minimum sales
Data Formats

• Start with files (not databases or data cubes)
• **File format** is a set of rules a file's structure should obey
• Common file formats for data:
  - Comma-separated values (CSV)
  - Tab-separated values (TSV) (or other delimiters)
  - Fixed-width format: old school
  - JavaScript Object Notation (JSON)
  - eXtensible Markup Language (XML)
Data Formats

• Start with files (not databases or data cubes)
• **File format** is a set of rules a file's structure should obey
• Common file formats for data:
  - Comma-separated values (CSV)
  - Tab-separated values (TSV) (or other **delimiters**)
  - Fixed-width format: old school
  - **JavaScript Object Notation (JSON)**
  - eXtensible Markup Language (XML)
Comma-separated values (CSV) Format

- Comma is a field separator, newlines denote records
  - a,b,c,d,message
  - 1,2,3,4,hello
  - 5,6,7,8,world
  - 9,10,11,12,foo

- May have a header (a,b,c,d,message), but not required

- No type information: we do not know what the columns are (numbers, strings, floating point, etc.)
  - Default: just keep everything as a string
  - Type inference: Figure out what type to make each column based on what they look like

- What about commas in a value? → double quotes
Delimiter-separated Values

• Comma is a **delimiter**, specifies boundary between fields
• Could be a tab, pipe (|), or perhaps spaces instead
• All of these follow similar styles to CSV
Fixed-width Format

- Old school
- Each field gets a certain number of spots in the file
- Example:

  - id8141  360.242940  149.910199  11950.7
  - id1594  444.953632  166.985655  11788.4
  - id1849  364.136849  183.628767  11806.2
  - id1230  413.836124  184.375703  11916.8
  - id1948  502.953953  173.237159  12468.3

- Specify exact character ranges for each field, e.g. 0-6 is the id
JavaScript Object Notation (JSON)

- A format for web data
- Associative arrays and lists
- Example:
  
  ```json
  { "name": "Wes", 
    "places_lived": ["United States", "Spain", "Germany"], 
    "pet": null, 
    "siblings": 
      [{ "name": "Scott", "age": 25, "pet": "Zuko"}, 
      { "name": "Katie", "age": 33, "pet": "Cisco"}]
  }
  
  Only contains literals (no variables) but allows null

  Values: strings, arrays, dictionaries, numbers, booleans, or null
  - Dictionary keys must be strings
  - Quotation marks help differentiate string or numeric values
eXtensible Markup Language (XML)

• Self-describing format with nesting
• Each field has tags
• Example:

  - <INDICATOR>
    <INDICATOR_SEQ>373889</INDICATOR_SEQ>
    <PARENT_SEQ></PARENT_SEQ>
    <AGENCY_NAME>Metro-North Railroad</AGENCY_NAME>
    <INDICATOR_NAME>Escalator Avail.</INDICATOR_NAME>
    <PERIOD_YEAR>2011</PERIOD_YEAR>
    <PERIOD_MONTH>12</PERIOD_MONTH>
    <CATEGORY>Service Indicators</CATEGORY>
    <FREQUENCY>M</FREQUENCY>
    <YTD_TARGET>97.00</YTD_TARGET>
  </INDICATOR>

• Top element is the root
Next

• Exploring Data & Visualization
• Install Tableau
  - Students receive a **free** license
  - [https://www.tableau.com/academic/students](https://www.tableau.com/academic/students)
• Install Anaconda (Python 3):
  - [https://www.anaconda.com/download/](https://www.anaconda.com/download/)
• Watch Tableau tutorials:
  - [https://www.tableau.com/learn/training](https://www.tableau.com/learn/training)
• Try "Hello World" in python
• Chapter 1 of *Python for Data Analysis* (see myCourses)