Advanced Data Management (CSCI 490/680)

Reproducibility

Dr. David Koop
Provenance in Computational Science

- Data Management
- Computation
- Visualization
- Provenance
- Publishing

Our research has been funded by the National Science Foundation and IIS.

Acknowledgments

References

The original workflow is available at: http://www.stccmop.org

Visualizing a binary star system simulation: This is an image that was generated by embedding a workflow diagram created using Vistrails workflow builder.

The VisTrails Project

Many sites for visualization at internet scale

The VisIt Visualization Tool

The application visualization system - computational environment for scientific visualization

Publication system - (omputational environment for scientific visualization

The design and realization of the virtual research environment for social sharing of workflows

Harnessing the Web Information Ecosystem with Wikiware Architectures

Visualizing a journal that serves the computational sciences (ommunity

Managing rapidly evolving scientific workflows

Towards provenance enabling paraview

Towards provenance enabling paraview

See: http://www.swivel.com

See: http://www.research.ibm.com/
Database Provenance

• Motivation: Data warehouses and curated databases
  - Lots of work
  - Provenance helps check correctness
  - Adds value to data by how it was obtained

• Three Types:
  - Why (Lineage): Associate each tuple t present in the output of a query with a set of tuples present in the input
  - How: Not just existence but routes from tuples to output (multiple contrib.'s)
  - Where: Location where data is copied from (may have choice of different tables)

[Cheney et al., 2007]
Why Provenance

<table>
<thead>
<tr>
<th>Agencies</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁: BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>t₂: HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExternalTours</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₃: BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>t₄: BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>t₅: BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>t₆: BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>t₇: HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>t₈: HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

Q₁:
SELECT a.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name AND e.type='boat'

Result of Q₁:
<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

- Lineage of (HarborCruz, 831-3000): {Agencies(t2), ExternalTours(t7)}
- Lineage of (BayTours, 415-1200): {Agencies(t1), ExternalTours(t5, t6)}
- This is not really precise because we don't need both t₅ and t₆—only one is ok

[Cheney et al., 2007]
How Provenance

- How provenance gives more detail about how the tuples provide witnesses to the result

- Prov of \((\text{San Francisco, 415-1200})\): \{\{t1\}, \{t1,t3\}\}

- \(t1\) contributes **twice**

- Uses provenance semirings (the "polynomial" shown on the right)

---

**Agencies**

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>t2</td>
<td>HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

**ExternalTours**

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>t4</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>t5</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>t6</td>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>t7</td>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>t8</td>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

\[Q_2:\]

```
SELECT e.destination, a.phone 
FROM Agencies a, 
(SELECT name, based_in AS destination 
FROM Agencies a 
UNION 
SELECT name, destination 
FROM ExternalTours e 
) t 
WHERE a.name = e.name 
```

**Result of \(Q_2\):**

<table>
<thead>
<tr>
<th>destination</th>
<th>phone</th>
<th>(t1 \cdot (t1 + t3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco</td>
<td>415-1200</td>
<td>(t1 + t2)</td>
</tr>
<tr>
<td>Santa Cruz</td>
<td>831-3000</td>
<td>(t1 \cdot t6)</td>
</tr>
<tr>
<td>Santa Cruz</td>
<td>415-1200</td>
<td>(t1 \cdot (t4 + t5))</td>
</tr>
<tr>
<td>Monterey</td>
<td>415-1200</td>
<td>(t1 \cdot t7)</td>
</tr>
<tr>
<td>Monterey</td>
<td>831-3000</td>
<td>(t1 \cdot t8)</td>
</tr>
<tr>
<td>Carmel</td>
<td>831-3000</td>
<td>(t1 \cdot t8)</td>
</tr>
</tbody>
</table>

---

[Cheney et al., 2007]
Where Provenance

- Where provenance traces to specific locations, not the tuple values
- Q and Q' give the same result but the name comes from different places
- Prov of HarborCruz in second output: (t2, name)
- Important in annotation-propogation

[Cheney et al., 2007]
VisTrails

- Comprehensive provenance infrastructure for computational tasks
- Focus on exploratory tasks such as simulation, visualization, and data analysis
- Transparently tracks provenance of the discovery process—from data acquisition to visualization
  - The trail followed as users generate and test hypotheses
  - Users can refer back to any point along this trail at any time
- Leverage provenance to streamline exploration
- Focus on usability—build tools for scientists
Version Trees for Evolution Provenance

- Undo/redo stacks are **linear**!
- We **lose history** of **exploration**
- Old Solution: User saves files/state
- VisTrails Solution:
  - **Automatically** & **transparently** capture entire history as a **tree**
  - Users can tag or annotate each version
  - Users can go back to **any** version by selecting it in the tree
Capturing Exploration: Version Tree of Workflows

1. Initial data
   - Corrected data
   - November ff
   - November 2 data
   - August 16 Tab

2. Station locations
   - Station map
   - Full fares map
   - Added fares

3. Difference
   - Broadway line
   - August 16
   - Broadway diff map

4. Sum of ffs
   - 30-D weekly
   - 161st-River

5. Concourse line
   - Filtered
   - Heatmap
Capturing Exploration: Version Tree of Workflows

- Initial data
  - Corrected data
    - November 1st
    - November 2 data
    - August 16 Tab
  - Sum of ffs
    - 30-D Weekly
    - 161st-River
  - Broadway line
  - Broadway Diff map
- Station locations
  - Station map
  - Added fares
    - Full fares map
    - August 16
    - Concourse line
      - Filtered
      - Heatmap
Workflow Upgrades
Workflow Upgrades
Workflow Upgrades

D. Koop, CSCI 680/490, Spring 2021
Workflow Upgrades
Querying Provenance by Example

- Provenance is represented as graphs: hard to specify queries using text!
- Querying workflows by example [Scheidegger et al., TVCG 2007; Beeri et al., VLDB 2006; Beeri et al. VLDB 2007]
  - WYSIWYG -- What You See Is What You Query
  - Interface to create workflow is same as to query
Assignment 5

- Four parts
  - Loading Data
  - Spatial Analysis
  - Graph Analysis
  - Temporal Analysis
- Due at the end of the semester (April 22, 2021)
- Questions?
Final Exam

- Monday, April 26, 4:00-5:50pm, Online (Blackboard)
- Similar format
- More comprehensive (questions from topics covered in Test 1 & 2)
- Will also have questions from temporal data, provenance, reproducibility, machine learning
- Page to be posted
Stronger Links Between Provenance and Data

- Filenames are often the mode of identification in data exploration
- We might also use URIs or access curated data stores
  - Always expected for exploratory tasks?
  - What happens if offline?
- Solution:
  - Managed store for data associated with computations
  - Improved data identification
  - Automatic versioning

[Koop et. al, 2010]
Provenance from Data

newfilename.dat

HASH CONTENTS

0ab678cd...

QUERY FILE STORE

FILE STORE

QUERY PROVENANCE

12ab3-45ef2...

INPUT REFERENCE

input files

12ab3-45ef2...

OBTAIEN INPUT FILES

OBTAIEN INPUT REFES

12ab3-45ef2...

QUERY FILE STORE

[Koop et. al, 2010]
Building Visualization Pipelines
Building Visualization Pipelines
Completions

[URL Completion, Safari]

[Code Completion, Intellisense]

[Web Search Completion, Google]
VisComplete Overview

- Mine provenance collection: Identify graph fragments that co-occur in a collection of workflows (Data-Driven)
- Predict sets of likely workflow additions to a given partial workflow
Suggestion Interface
Suggestion Interface
VisComplete Results
Visualization by Analogy
Visualization by Analogy
Visualization by Analogy
Visualization by Analogy
Generating Visualizations by Analogy

A is to B as C is to D
Generating Visualizations by Analogy

A is to B as C is to D.
Generating Visualizations by Analogy

- Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
Generating Visualizations by Analogy

- Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
- Find map between $A$ & $C$: $\text{map}(A,C)$
Generating Visualizations by Analogy

- Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
- Find map between $A$ & $C$: $\text{map}(A,C)$
- Compute mapped difference $\Delta AC(A,B) = \text{map}(A,C) \Delta(A,B)$
  - $D = \Delta AC(A,B) \circ C$
VisMashup

Acquire and Analyze Pipelines → Create Views (Simplify Pipelines) → Combine Views → App generation and deployment
VisTrails for Teaching Scientific Visualization

• “Using VisTrails and Provenance for Teaching Scientific Visualization” [Silva et al., Eurographics Educator Program, 2010]

• Same features that scientists use for exploratory tasks can also benefit students
  - Exploration: see all pipelines not just a “final” one
  - Comparison: see different pipelines and what changes exist
  - Assessment: see how a solution was developed
Provenance Analysis of Projects

Figure we computed these percentages for each task, as shown in and layout actions across all tasks (Figure types of actions involved. For all users, we calculated the the locations of modules in visual programming interface). parameter actions in the workflow); and

actions involved in workflow development into: different types of work involved in a task, we classified the
certainty, due date, and how open-ended they were. To illustrate
categorize tasks. As noted in Table

Workflow evolution information can also be helpful to char-

character on the user; they can do their work without caring about
VisTrails captures provenance is that there is no extra bur-

One of the really nice features of the unobtrusive way that

approached by someone with more experience. When mak-

makes it possible for the instructors to share their own work

formation, it is possible for one person to see what another

user's learning experience. Due to the provenance in-

tions of provenance-enabled systems. Provenance inform-

Branching is just one variable from the workflow evolu-

The Eurographics Association 2010.
Provenance Analysis of Projects

Comparing Paths to Solutions for Two Students
The State of Repeatability in Computer Systems Research

C. Collberg and T. Proebsting
CACM 2016
State of Repeatability in Computer Systems

- "Cool paper! Can you send me the system?"
- How hard is it to just re-execute published experiments
- Most people say they will share their code and data are available…
- Weak repeatability: Do authors make the source code used to create the results in their article available, and will it build?
Figure 4: Process by which the study was performed.

[Collberg and Proebsting, 2015]
Figure 11: Study result. Blue numbers represent papers that were excluded from consideration, green numbers papers that are weakly repeatable, red numbers papers that are non-weakly repeatable, and orange numbers represent papers that were excluded (due to our restriction of sending at most one email to each author).

D. Koop, CSCI 680/490, Spring 2021
Excuses

• "Unfortunately the current system is not mature"
• "The code was never intended to be released so it is not in any shape for general use"
• "[Our] prototype included many moving pieces that only [student] knew how to operate... he left"
• "... the server in which my implementation was stored had a disk crash ... three disks crashed... Sorry for that"

[Collberg and Proebsting, 2015]
Excuses

• "…when we attempted to share it, we [spent] more time getting outsiders up to speed than on our own research"
• "… we can't share what [we] did for this paper. … this is not in the academic tradition, but this is a hazard in an industrial lab"
• "… based on earlier (bad) experience, we [want] to make sure that our implementation is not used in situations that it is not meant for"

[Collberg and Proebsting, 2015]
Excuse Classification

• Versioning
• Available Soon
• No Intention to Share
• Personnel Issues
• Lost Code
• Academic Tradeoffs
• Industrial Lab Tradeoffs
• Obsolete HW/SW
• Controlled Usage
• Privacy/Security
• Design Issues

[Collberg and Proebsting, 2015]
Some of these are (partially) people problems, not technical problems
Examining 'Reproducibility in Computer Science'

- Repeat the experiment in reproducibility!
- Differences from original
- Shows issues with trying to classify experiments

### Progress

- Purported Not Building; Disputed; Not Checked: 6%
- Purported Building; Disputed; Not Checked: 2%
- Conflicting Checks!: 0%
- Misclassified: 1%
- Purported Not Building But Found Building: 14%
- Purported Building But Found Not Building: 0%
- Purported Not Building; Confirmed: 0%
- Purported Building; Confirmed: 0%
- All Others Purported Not: 27%
Recommendations

- Fund repeatability engineering
- Require sharing contracts

<table>
<thead>
<tr>
<th>Location</th>
<th>• email address and/or web site</th>
</tr>
</thead>
</table>
| Resource       | • **types:** code, data, media, documentation  
|                | • **availability:** no access, access, NDA access  
|                | • **expense:** free, non-free, free for academics  
|                | • **distribution form:** source, binary, service  
|                | • **expiration date**  
|                | • **license**  
|                | • **comment**  
| Support        | • **kinds:** resolve installation issues, fix bugs, upgrade to new language and operating system versions, port to new environments, improve performance, add features  
|                | • **expense:** free, non-free, free for academics  
|                | • **expiration date**  

[Collberg and Proebsting, 2015]
Reproducible Research

• Science is verified by replicating work independently
• Replication Issues:
  - Requires many resources to replicate (Sloan Digital Sky Survey)
  - Requires significant computing power (Climate Model Simulation)
  - Requires too much time or very specific circumstances (Environment Epidemiology)
• Reproducibility
  - Replication of the analysis based on the collected data (not replicating the data collection itself)
  - Better if we have the actual code or available executables
Reproducibility Spectrum

Publication only

Publication +

Code

Code and data

Linked and executable code and data

Full replication

Gold standard

Not reproducible
Published Papers

• “It’s impossible to verify most of the results that computational scientists present at conference and in papers.” [Donoho et al., 2009]

• “Scientific and mathematical journals are filled with pretty pictures of computational experiments that the reader has no hope of repeating.” [LeVeque, 2009]

• “Published documents are merely the advertisement of scholarship whereas the computer programs, input data, parameter values, etc. embody the scholarship itself.” [Schwab et al., 2007]
Problem: Incomplete Publications

- A paper cannot include all relevant details of the science
  - Large volumes of data
  - Complex processes
  - Code dependencies

- This makes publishing complete results more difficult!
Reproducible/Executable Papers
Reproducible/Executable Papers

![Diagram](image-url)

**Figure 1:** The Weka's original screenshot displays information about the dataset. The package provides a graphical user interface (GUI) that allows the user to interact with the data. The application offers various tools for data exploration, including visualization tools. The GUI displays various statistics about the dataset, such as the number of instances and attributes. Clicking on the "datasets" or the "weka" tab provides detailed information about the dataset selected by the user. The app allows for easy manipulation and analysis of the data.

**Level 3 Enhancements**

As shown in the 3D visualization of the Weka's original screenshot, the application provides advanced features, including interactive data exploration tools. These features allow users to manipulate data and visualize patterns in real-time. The GUI is designed to be user-friendly, allowing for quick data analysis and visualization.

---

**References**

Koop, D., CSCI 680/490, Spring 2021

---

For more information on the original simulation model, please refer to the paper "Weka's Original Simulation Model" available in the course resources. The paper details the development and implementation of the simulation model in Weka, focusing on its capabilities and potential applications in various fields.
Reproducible/Executable Papers

Figure 2. The WaterTouch window that allows users to select the “Figure 2” file associated with the original article. To generate the animation, the user first needs to select the file name, then choose the appropriate file format (Movie, PNG, Gif, etc.), and finally click on the “Export” button. The WaterTouch application then renders the animation frame by frame, using the original code and parameters from the original article.

As a VTK file, the software parameter file contains all the necessary information to reproduce the original visualization accurately.

Level 3 Enhancements

As the example at www.reproducibility.org shows, our VTK-based visualization tools offer a significant enhancement over traditional journal articles. By clicking the red “Download” button displayed within the article, readers can access the VTK code and replicate the figures and results presented in the original article. The VTK code is open-source and freely available, allowing researchers and analysts to build upon and extend the work described in the original article. This not only enhances the reproducibility of the research but also facilitates further exploration and innovation in the field.
Reproducible/Executable Papers

as a VisTrails workflow process in Figure 3. Refer to the original article, placing together a visualization workflow that is essentially an underlying representation of the data that matches our conceptual fluid streamlines. It's unusual for computer science students to invent such a visualization workflow. Typically, they use concepts from the field to visualize fluid streamlines. In the original article, the authors developed a conceptual visualization workflow to invent such a representation. This is unusual for computer science students to invent such a visualization workflow. Typically, they use concepts from the field to visualize fluid streamlines.
Reproducible/Executable Papers

This level of enhancement leverages a powerful tool to visualize and analyze the underlying properties of the data that are not visually apparent. By using a computational fluid dynamics (CFD) model, we can visualize the scientific results of this study in two different views (Figure 3). This CFD model was developed using the VisTrails App, which integrates various visualization tools into a single environment. This approach allows for a more intuitive understanding of the data and facilitates the development of new insights. As shown in Figure 4, the CFD model is used to simulate the flow behavior in a real-world scenario, providing a detailed view of the fluid dynamics at different locations.

To further enhance the visual representation, a new parameter is introduced. This parameter, labeled as "speed," plays a crucial role in adjusting the visualization. By modifying the "speed" parameter, we can observe how changes in this parameter affect the overall visualization. This feature enables users to explore the data in a more dynamic and interactive manner, thus enhancing the overall understanding of the complex phenomena being studied.

In conclusion, the use of computational fluid dynamics and advanced visualization tools has significantly enhanced our ability to analyze and interpret complex data sets. These tools not only provide a more intuitive understanding of the underlying processes but also facilitate the development of new hypotheses and insights. As we continue to refine and improve these tools, we can expect to see even more significant advancements in the field of computational science.
Challenges

• Re-using results
• Adding results to publications
• Obtaining results, computations, and input from publications
• Publishing interactive experiments
• Searching executable paper collections
• Reviewers: execution environments, checking different parameters
• Longevity/maintenance
• Resource constraints:
  - analyses run on supercomputers
  - large datasets
  - privacy or intellectual property concerns
General Strategies for Reproducibility

• Preserving the Mess:
  - Just save a virtual machine
  - Trace dependencies

• Encouraging Cleanliness:
  - Use a system (e.g. Umbrella, VisTrails)
  - Use literate programming environments
  - Use code and data repositories
  - Use packaging system (ReproZip)

[Categories from H. Meng et al., 2016]
Literate Programming

- Knuth’s WEB system
- Mathematica
- Code this is well-documented using comments
- Jupyter Notebooks
Data and Code Availability

• Code Repositories:
  - GitHub
  - GitLab
  - ...

• Data Repositories:
  - figshare, freebase, dryad, DataONE
  - Also many domain-specific repositories
  - http://oad.simmons.edu/oadwiki/Data_repositories
10 Rules for Reproducible Computational Research

• Rule 1: For Every Result, Keep Track of How It Was Produced
• Rule 2: Avoid Manual Data Manipulation Steps
• Rule 3: Archive the Exact Versions of All External Programs Used
• Rule 4: Version Control All Custom Scripts
• Rule 5: Record All Intermediate Results, When Possible in Standardized Formats

[Sandve et al., 2013]
10 Rules for Reproducible Computational Research

- Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
- Rule 7: Always Store Raw Data behind Plots
- Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
- Rule 9: Connect Textual Statements to Underlying Results
- Rule 10: Provide Public Access to Scripts, Runs, and Results

[Sandve et al., 2013]
Rules or Benefits?

- Laws to make sure people don't cheat or lie or steal
- Is that a good incentive? You won't be mislabeled as a criminal?
- Benefits of Reproducibility
  - Reproducible programs can be compared
  - Reproducible software and results are documented
  - Reproducible software is portable
  - Reproducible experiments are cited

[J. Freire et al.]
Reproducible Experiments Classification

- Depth: how much is available?
  - figures
  - scripts
  - raw data
  - experiments
  - software system

- Portability: what machine specs are necessary?
  - same machine
  - similar machine
  - different OS

- Coverage: how much can be reproduced?
(Database) Research Topics

- Design and Management of Experiment Repositories
- Querying and Searching Experiments
- Mining Experiments
A Large-scale Study about Quality and Reproducibility of Jupyter Notebooks

J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire
Notebooks and Hidden State

Fibonacci

```python
def fib(x):
    if x <= 1:
        return x
    return fib(x-1) + fib(x-2)

fib(10)
```

Let's plot the numbers

```python
from matplotlib import pyplot

x = range(15)
y = [fib(n) for n in x]
pyplot.plot(x, y);
```

Output 1

![Output 1](image)

Output 2

![Output 2](image)

In [1]: co = 0
In [1]: co = 0
In [1]: co = 0

In [3]: co += 1
In [2]: co += 2
In [3]: co
In [3]: co

Out[4]: 2
Out[3]: 1
Out[3]: 1

[Pimentel et al., 2019]
Notebook Composition

![Graph showing the composition of notebooks]

[D. Koop, CSCI 680/490, Spring 2021]
Notebook Reproducibility

- Use notebooks from Github (~1 million)
  - Unambiguous cell order? 81.99%
- Study notebook dependencies
  - Dependencies Available? 13.72%
  - Dependencies Install? 5.03%
- Study notebook executability
  - Execute: 24.11% of unambiguous cell order
  - Matched results: 4.03%

[Pimentel et al., 2019]
Best Practices

• Use short titles with a restrict charset (A-Z a-z 0-9 . -) for notebook files and markdown headings for more detailed ones in the body
• Pay attention to the bottom of the notebook. Check whether it can benefit from descriptive markdown cells or can have code cells executed or removed
• Abstract code into functions, classes, and modules and test them
• Declare the dependencies in requirement files & pin versions of all packages
• Use a clean environment to test if dependencies are properly declared
• Put imports at the beginning of notebooks
• Use relative paths for accessing data in the repository
• Re-run notebooks top to bottom before committing

[Pimentel et al., 2019]
Problem: What is df at any point in time?
Dataflow Notebooks: Resolve Notebook Ambiguities

In [d51f8eab]:
```python
import pandas as pd
df = pd.read_csv('guardian-top100-female-2019.csv')
```

In [full]:
```python
df = df.rename(columns={'Age on 1 Dec 2019': 'Age'})
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age on 1 Dec 2019</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam Kerr</td>
<td>1</td>
<td>Forward</td>
<td>26</td>
<td>Australia</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ludmila</td>
<td>100</td>
<td>Forward</td>
<td>25</td>
<td>Brazil</td>
</tr>
</tbody>
</table>

100 rows x 5 columns

In [over30]:
```python
df = df[df.Age >= 31]
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Megan Rapinoe</td>
<td>3</td>
<td>Midfielder</td>
<td>34</td>
<td>USA</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Cláudia Neto</td>
<td>97</td>
<td>Midfielder</td>
<td>31</td>
<td>Portugal</td>
</tr>
</tbody>
</table>

19 rows x 5 columns

In [under25]:
```python
df = df[df.Age <= 24]
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Position</th>
<th>Age</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada Hegerberg</td>
<td>4</td>
<td>Forward</td>
<td>24</td>
<td>Norway</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Lena Oberdorf</td>
<td>99</td>
<td>Midfielder</td>
<td>17</td>
<td>Germany</td>
</tr>
</tbody>
</table>

25 rows x 5 columns
Dataflow Notebooks: Dependency Graph

- Shows connections between cells
- Can see which cells would be affected by a change
- Same colors indicate which parts of the graph are stale
- Linked to the notebook
  - Hover to show a cell's code
  - Can also execute in the graph