Sharing data pipelines

Why sharing data may not be enough, and what to do about it

David Käthner
Deutsches Zentrum für Luft- und Raumfahrt e.V.
07.12.2020
Three claims about data pipelines

1. Providing only primary data is often insufficient to make results of an analysis sufficiently transparent.

2. Data pipelines and data analysis are inseparably linked.

3. The idea of data pipelines is easy to understand, but can be difficult to implement.
Data Pipelines have a direct effect on reproducibility

Research question (actual study):
Is a certain pattern in EEG-profiles indicative of a diagnosis from the autism spectrum disorder (ASD)?

Answer:
Not sure, because very different outcomes using the same data. Reason for that: Different data pipelines involving different software.

Why worry about pipelines?

Bugs

Pandas reads values incorrectly from csv

Features

Scientists rename human genes to stop Microsoft Excel from misreading them as dates

Sometimes it’s easier to rewrite genetics than update Excel

By James Vincent | Aug 6, 2020, 8:44am EDT

Parameter settings

```python
# Filtering
data.filter(phas='zero',
    fir_window='hamming',
    fir_design='firwin',
    l_freq=None, h_freq=48,
    l_trans_bandwidth='auto',
    h_trans_bandwidth='auto',
    filter_length='auto')
```
You have always been creating data pipelines

Research question:
Is a certain pattern in EEG-profiles indicative of a diagnosis from the autism spectrum disorder (ASD)?

<table>
<thead>
<tr>
<th>ID</th>
<th>values</th>
<th>ID</th>
<th>values</th>
<th>ID</th>
<th>time</th>
<th>values (raw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>02AN</td>
<td>...</td>
<td>02AN</td>
<td>...</td>
<td>02AN</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>95BC</td>
<td>...</td>
<td>95BC</td>
<td>...</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20NQ</td>
<td>...</td>
<td>20NQ</td>
<td>...</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

source 1: socio-demographic

source 2: diagnosis

source 3: EEG

join

<table>
<thead>
<tr>
<th>ID</th>
<th>socdem</th>
<th>diagnosis</th>
<th>EEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>02AN</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>95BC</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20NQ</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
1NF-normalization, also known as Tidy Data

Each row = one observation
- uniquely identifiable through one or more keys
- observation ≠ point in time

Each column = one variable
- no further decomposition possible
- is of one of these types:
  - identification (key)
  - factor
  - measurement
  - factor from measurement

Regression:

\[ y = x_0 + x_1 + x_2 \]

Normalization: [link](#)

Tidy Data: Wickham, H. (2014)
Functions are the core concept of a working pipeline

<table>
<thead>
<tr>
<th>ID</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>source 1</td>
</tr>
<tr>
<td>2</td>
<td>source 1</td>
</tr>
<tr>
<td>3</td>
<td>source 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>source 2</td>
</tr>
<tr>
<td>2</td>
<td>source 2</td>
</tr>
<tr>
<td>3</td>
<td>source 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>source 3</td>
</tr>
<tr>
<td>2</td>
<td>source 3</td>
</tr>
<tr>
<td>3</td>
<td>source 3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Application of functions to data:

```
for data in source:
    for participant in ID:
        read_data() |>
        function_1() |>
        function_2() |>
        save_data()  
```

Pipe operator:

```
output function 1 = input function 2
```

Extract   Transform   Load

Read      Rename      Save

denoise aggregate

...
R and Tidyverse: Guiding you in the right direction

- "opinionated collection of R packages"
- addresses every important aspect of data preparation and analysis specifically
- accessible tutorials and documentation
- create literal data pipelines

```
country  beer_servings spirit_servings wine_servings total_litres_of_pure_alcohol
1 Afghanistan 0 0 0 0
2 Albania  89 132 54 4.9
3 Algeria  25 0 14 0.7
```

```
drinks_smaller <- drinks %>%
  filter(country %in% c("USA", "China", "Italy", "Saudi Arabia")) %>%
  select(-total_litres_of_pure_alcohol) %>%
  rename(beer = beer_servings, spirit = spirit_servings, wine = wine_servings)
```

```
country  beer  spirit  wine
1 China  79   192    8
2 Italy  85   42    237
3 Saudi Arabia  0   5      0
4 USA  249  158    84
```
KNIME (Konstanz Information Miner): Taking pipeline literally

- each node is a processing step
- nodes can be configured via GUI or call a script (R, Python, Matlab)

personal opinion:
- usage requires programming skills
- really good with pipelines that are much re-used
- visual approach works for simpler workflows, complex tasks still need to be scripted
- lots of machine learning for finance and life sciences
Sharing *reproducible* pipelines

Pipeline must be deterministic

- same input = same output
  - most likely problem in EEG-parameter extraction: dependency issue (e.g. changed defaults)
- 2 conditions
  - ensure actually same input is used
  - solve dependencies

How to

1. make sure the right data is used
   - provide a hash value for your data like so:
     ```
     $ tar -cf - raw | md5sum
     9d8a7c7dc804237438252af02374065d4
     ```
2. reserve the dependencies
   - R and Packrat: provide project containing all dependencies as local copies

more options

- R: checkpoint
- Python: virtual environments
- everything: Docker container
Tips for a shareable data pipeline

practicalities

• isolate functions in separate file
• load functions into cleaning file, notebook, ...
• ship files to others via Github, servers of your organization, Email, ...

make your own life easier

• design the pipeline before implementation
  • better even before data recording
  • best to already know planned analysis
• adopt a functional programming style (40-line rule-of-thumb)
• be very careful with interactive programming (best to avoid it for cleaning)

improve usability for others

• can be executed without (external) explanation
  • README is a good starting point
• group steps of cleaning/analysis in coherent units that communicate intent
• document settings in an obvious way (e.g. config-files)

SPSS

• try to do as much as possible in SPSS, and export the syntax file
Three claims about data pipelines revisited

1. Providing only primary data is often insufficient to make results of an analysis sufficiently transparent.
   - bugs, features, parameter settings, user error, missing or changing dependencies

2. Data pipelines and data analysis are inseparably linked.
   - basis for many types of analysis, visualisations, ... are tidy data

3. The idea of data pipelines is easy to understand, but can be difficult to implement.
   - Good: Effective and accessible concepts, tools, and tutorials exist.
   - Bad: Complex pipelines still require substantial programming and experience.
   - Solution: Teach basic concepts and their application as part of regular curriculum.
Thank you for your attention!
References


Websources

Database normalization
https://de.wikipedia.org/wiki/Normalisierung_(Datenbank)

Docker
https://www.docker.com/resources/what-container
https://www.docker.com/blog/containerized-python-development-part-1/

Packrat
https://rstudio.github.io/packrat/
https://github.com/c06n/Packrat_HowTo

Pandas issue
https://github.com/pandas-dev/pandas/issues/34120
https://github.com/c06n/Pandas_readcsv_issue/blob/master/demonstration.ipynb

Tidyverse
https://moderndive.com/index.html
https://r4ds.had.co.nz/

KNIME
https://www.knime.com/learning
Different pipelines, different Alpha peak and Alpha Frequencies

Data Pipeline of Lefebvre et al. (2018)

load and transform log data

denoising

join

provide output

compute factor variables from data

FIGURE S2 | Preprocessing and processing chain of the data.
**Big data and data models**

<table>
<thead>
<tr>
<th>target</th>
<th>source</th>
<th>typical frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>eye tracking</td>
<td>30 / 120 / 500 / 1000 Hz</td>
</tr>
<tr>
<td></td>
<td>ECG</td>
<td>500 / 1000 Hz</td>
</tr>
<tr>
<td>context</td>
<td>vehicle data</td>
<td>20 / 25 Hz</td>
</tr>
<tr>
<td></td>
<td>positional data</td>
<td>1 Hz</td>
</tr>
</tbody>
</table>

back-of-the-envelope calculation for logging at 120 Hz:
- 100 variables, float64, 30 min, 50 participants = 8.64 GB

• **Big data problem**
  - per participant easily 2-5 GB
  - *all* recorded data does not fit in standard working memory anymore

• **Ontological problem**
  - tidy model is intuitive, but forcing every data set into this model leads to inefficient data structures
  - error prone for ontologically different dimensions
NucleiClassification
* example data can be found under the link in the description.
* select images with the two row filters in order to train a classifier. At the moment, the first row filter selects a certain subposition, the second a specific row.
* open the Labeling Editor and defines a training set by creating new classes and clicking on the according objects in the image. Refer to the help of the node for details.
* run the rest of the workflow to classify the whole dataset.
* configure the Excel Writer to save your data accordingly.

Complex KNIME pipeline

If you like this workflow and want to acknowledge it in your work, please do so! This workflow was composed by Manuel Günkele (manuel.gunkele@bioquant.uni-heidelberg.de) as part of the German Node of the Bioinformatics infrastructure (HDMI) as part of the NIH grant 5U54HG006135.

Please consider visiting https://www.de.nbi.de for more information and interesting tools and services!

Please consider also giving us feedback via the SurveyMonkey link in the description, or visit https://www.surveymonkey.de/r/dbs-service?/sc/hub/feedbackde to do so. It's really really short just three questions, remarks and comments are optional.

Many Thanks!

link