MY472 – Week 1: Overview and Fundamentals

Friedrich Geiecke

MY472: Data for Data Scientists

27 September 2021

Course website: lse-my472.github.io
What is this course about?
The 80/20 rule of data science:
80% data manipulation, 20% data analysis

It is about the 80%
In more detail

Course tries to provide “data science literacy”

- **What** is data?
  - Basic data types and structures
- **How to collect** data?
  - How to scrape data from the internet
  - How to work with APIs
- **How to clean and process** data?
  - How to format, organize, and reshape data
  - Cloud computing to process very large datasets
- **How to store and query** data?
  - How to create and use databases
  - How to create and manage (online) databases
Tools applicable in a wide range of fields

For example

- Private sector
- Public sector
- Healthcare
- Non-profit
- Data journalism
- Research
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- Markdown in brief
- git and Github for version control
Course outline

1. Introduction to data
2. The shape of data
3. HTML and CSS
4. Using data from the internet
5. Working with APIs
6. (Reading week)
7. Textual data
8. Data visualisation
9. Creating and managing databases
10. Interacting with online databases
11. Cloud computing
Prerequisites and software

- Introductory course – no prerequisites (only completion of R preparatory course required!)
- Lab computers are available, but we strongly recommend bringing your own laptop
- Software:
  - R 4.1.1 – Install from https://www.r-project.org/
  - RStudio – Install from https://www.rstudio.com/products/rstudio/download/
  - GitHub Desktop – Install from https://desktop.github.com/
    → Please install before lab session this week
- Mirrors similar tool usage and learning in other Methodology courses
About me

▶ Assistant Professor of Computational Social Science at the London School of Economics
  ▶ PhD in Economics, London School of Economics, 2020

▶ Research:
  ▶ Topics at the boundary of machine learning and economics
  ▶ Innovation, macroeconomic fluctuations, economic policy
  ▶ Natural language processing, reinforcement learning, statistical machine learning

▶ Contact:
  ▶ f.c.geiecke@lse.ac.uk
  ▶ https://sites.google.com/view/friedrichgeiecke/
Your turn

1. Name?
2. MSc/PhD Programme?
3. Previous experience with R?
4. Why are you interested in this course and what would you most like to learn?
Course philosophy

How to learn the techniques in this course?

▶ Lecture approach: not ideal for learning how to code
▶ You can only learn by doing
→ We will cover each concept three times during each week
   1. Introduction to the topic in lecture
   2. Guided coding session in lecture and lab
   3. Course assignments
▶ We will move relatively fast
Course webpage: https://lse-my472.github.io/

- Mixed set of readings, very specific to each week
  - Often freely available online, otherwise, available for purchase (often in electronic versions)
  - Some books are (freely) available online and in print, and the online version may be more recent

- Please do the readings!
Course meetings

- Pre-recorded lectures
- One-hour lecture discussions (also called ’Q&A’) via Zoom (you only have to attend one per week)
  - Group 1: Tuesdays 09:00–10:00 via Zoom
  - Group 2: Tuesdays 15:00–16:00 via Zoom
- Ten one-hour classes (“labs”)
  - Group 1: Fridays 11:00–12:00 in KSW.1.01 and via Zoom
  - Group 2: Fridays 16:00–17:00 in NAB.2.04 and via Zoom
- No lecture/class in Week 6
- Office hours (book via StudentHub)
  - Friedrich: Tuesdays 16:00-18:00
  - Patrick: Tuesdays 14:00-15:00 and Fridays 15:00-16:00
Assessment

- 5 assignments will be assessed (50%).
  - Submitted via GitHub (more in lab)
  - Only “knitted” R-markdown assignments in HTML accepted
  - One will be collaborative; rest will be individual submissions
- Take-home assignment (50%)
  - Individual assignment that asks to answer a series of questions with data
  - More open-ended format than assignments
  - Deadline: 14 January 2022, 14:00
A note on plagiarism and collaboration

- Four individual term time assignments, one group term time assignment, one individual final assignment
- **Strictly no discussion and collaboration with others allowed in any individual assignment**
- You can use online resources but always give credit and cite if you borrow code or solutions
- Any forbidden collaboration or not cited code/solutions/papers/resources are considered plagiarism
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- Markdown in brief
- git and GitHub for version control
History of data

Rosetta Stone, British Museum

- Great book on the history of information and data: The Information by James Gleick (not on the formal reading list)
- Early example of database often government records: Who is paying taxes and how much, census of citizens, etc.
Early example of a database index

Index cards used in a library catalog of books

- Initially developed to catalog species by botanist Carl Linnaeus (19th century)
- Units (species, books) are a record; records are indexed using a specific reference / sorting system
Each record looked like this:
Dewey decimal system

- A proprietary library classification system first published in the United States by Melvil Dewey in 1876
- Scheme is made up of ten classes, each divided into ten divisions, each having ten sections
- The system’s notation uses Arabic numbers, with three whole numbers making up the main classes and sub-classes and decimals creating further divisions
- Example:
  - 500 Natural sciences and mathematics
    - 510 Mathematics
    - 516 Geometry
      - 516.3 Analytic geometries
      - 516.37 Metric differential geometries
      - 516.375 Finsler Geometry
Problem: Cards only sorted in one way. Re-referencing literally a manual operation

Contrast with the idea of electronic indices, where assets are stored once and many indexing and referencing systems can be applied
Relational databases

- Codd, E.F. (1970) “A Relational Model of Data for Large Shared Data Banks.” *Communications of the ACM.*
Recent developments in data storage/management

- **NoSQL**: beyond relational structure; flexible; more scalable & compatible with distributed cloud storage (Big Data)
Trying to define Big Data

1. **Volume**: Around 8 billion mobile phones, around 2.5 billion Facebook users, 500+ million tweets per day...

2. **Velocity**: How quickly is data flowing? Personal, spatial and temporal granularity

3. **Variability**: Images, networks, long and short text, geographic coordinates, streaming...

Dumbill (2012), Monroe (2013)

**Big Data**: Data that is so large, complex, and/or variable that some new tools to understand it must be created
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- Markdown in brief
- git and Github for version control
Changes in the world of data

- Volume of data in the modern world: Very large fraction of the world’s data has been generated in the last *two years*
- Facebook processes 500+ terabytes of data each day
- Square Kilometer Array (SKA) telescope
  - Southern hemisphere radio telescope with a total of 1km² of data sensors
  - Will generate 1 exabyte *daily* = $10^{18}$ bytes
Compare this with the Apollo Guidance Computer (1966), which guided the first humans to the moon:

- Magnetic core memory: 16-bit word length, 2048 words RAM = 4KB
- Core rope memory: 36,864 words. 73KB
Basic units of data

- **Bits**
  - Smallest unit of storage; a 0 or 1
  - With $n$ bits, can store $2^n$ patterns

- **Bytes**
  - 8 bits = 1 byte (why 1 byte can store 256 patterns)
  - “eight bit encoding” - used to represent characters, such as A represented as 65 = 01000001
<table>
<thead>
<tr>
<th>Dec</th>
<th>Hx</th>
<th>Oct</th>
<th>Char</th>
<th>ASCII</th>
<th>HTML</th>
<th>Char</th>
<th>ASCII</th>
<th>HTML</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>000</td>
<td>NUL (null)</td>
<td>32</td>
<td>0x20</td>
<td>Space</td>
<td>64</td>
<td>0x40</td>
<td>Space</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>001</td>
<td>SOH (start of heading)</td>
<td>33</td>
<td>0x21</td>
<td>!</td>
<td>65</td>
<td>0x41</td>
<td>!</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>002</td>
<td>STX (start of text)</td>
<td>34</td>
<td>0x22</td>
<td>&quot;</td>
<td>66</td>
<td>0x42</td>
<td>&quot;</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>003</td>
<td>ETX (end of text)</td>
<td>35</td>
<td>0x23</td>
<td>#</td>
<td>67</td>
<td>0x43</td>
<td>#</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>004</td>
<td>EOT (end of transmission)</td>
<td>36</td>
<td>0x24</td>
<td>$</td>
<td>68</td>
<td>0x44</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>005</td>
<td>ENQ (enquiry)</td>
<td>37</td>
<td>0x25</td>
<td>%</td>
<td>69</td>
<td>0x45</td>
<td>%</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>006</td>
<td>ACK (acknowledge)</td>
<td>38</td>
<td>0x26</td>
<td>&amp;</td>
<td>70</td>
<td>0x46</td>
<td>&amp;</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>007</td>
<td>BEL (bell)</td>
<td>39</td>
<td>0x27</td>
<td>'</td>
<td>71</td>
<td>0x47</td>
<td>'</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>010</td>
<td>BS (backspace)</td>
<td>40</td>
<td>0x28</td>
<td>(</td>
<td>72</td>
<td>0x48</td>
<td>(</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>011</td>
<td>TAB (horizontal tab)</td>
<td>41</td>
<td>0x29</td>
<td>)</td>
<td>73</td>
<td>0x49</td>
<td>)</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>012</td>
<td>LF (NL line feed, new line)</td>
<td>42</td>
<td>0x2A</td>
<td>*</td>
<td>74</td>
<td>0x4A</td>
<td>*</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>013</td>
<td>VT (vertical tab)</td>
<td>43</td>
<td>0x2B</td>
<td>+</td>
<td>75</td>
<td>0x4B</td>
<td>+</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>014</td>
<td>FF (NP form feed, new page)</td>
<td>44</td>
<td>0x2C</td>
<td>,</td>
<td>76</td>
<td>0x4C</td>
<td>,</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>015</td>
<td>CR (carriage return)</td>
<td>45</td>
<td>0x2D</td>
<td>-</td>
<td>77</td>
<td>0x4D</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>016</td>
<td>SO (shift out)</td>
<td>46</td>
<td>0x2E</td>
<td>.</td>
<td>78</td>
<td>0x4E</td>
<td>.</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>017</td>
<td>SI (shift in)</td>
<td>47</td>
<td>0x2F</td>
<td>/</td>
<td>79</td>
<td>0x4F</td>
<td>/</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>020</td>
<td>DLE (data link escape)</td>
<td>48</td>
<td>0x30</td>
<td>0</td>
<td>80</td>
<td>0x30</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>021</td>
<td>DC1 (device control 1)</td>
<td>49</td>
<td>0x31</td>
<td>1</td>
<td>81</td>
<td>0x31</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>022</td>
<td>DC2 (device control 2)</td>
<td>50</td>
<td>0x32</td>
<td>2</td>
<td>82</td>
<td>0x32</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>023</td>
<td>DC3 (device control 3)</td>
<td>51</td>
<td>0x33</td>
<td>3</td>
<td>83</td>
<td>0x33</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>024</td>
<td>DC4 (device control 4)</td>
<td>52</td>
<td>0x34</td>
<td>4</td>
<td>84</td>
<td>0x34</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>025</td>
<td>NAK (negative acknowledge)</td>
<td>53</td>
<td>0x35</td>
<td>5</td>
<td>85</td>
<td>0x35</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>026</td>
<td>SYN (synchronous idle)</td>
<td>54</td>
<td>0x36</td>
<td>6</td>
<td>86</td>
<td>0x36</td>
<td>6</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>027</td>
<td>ETB (end of trans. block)</td>
<td>55</td>
<td>0x37</td>
<td>7</td>
<td>87</td>
<td>0x37</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>030</td>
<td>CAN (cancel)</td>
<td>56</td>
<td>0x38</td>
<td>8</td>
<td>88</td>
<td>0x38</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>031</td>
<td>EM (end of medium)</td>
<td>57</td>
<td>0x39</td>
<td>9</td>
<td>89</td>
<td>0x39</td>
<td>9</td>
</tr>
<tr>
<td>26</td>
<td>IA</td>
<td>032</td>
<td>SUB (substitute)</td>
<td>58</td>
<td>0x3A</td>
<td>:</td>
<td>90</td>
<td>0x3A</td>
<td>:</td>
</tr>
<tr>
<td>27</td>
<td>IB</td>
<td>033</td>
<td>ESC (escape)</td>
<td>59</td>
<td>0x3B</td>
<td>;</td>
<td>91</td>
<td>0x3B</td>
<td>;</td>
</tr>
<tr>
<td>28</td>
<td>IC</td>
<td>034</td>
<td>FS (file separator)</td>
<td>60</td>
<td>0x3C</td>
<td>&lt;</td>
<td>92</td>
<td>0x3C</td>
<td>&lt;</td>
</tr>
<tr>
<td>29</td>
<td>ID</td>
<td>035</td>
<td>GS (group separator)</td>
<td>61</td>
<td>0x3D</td>
<td>=</td>
<td>93</td>
<td>0x3D</td>
<td>=</td>
</tr>
<tr>
<td>30</td>
<td>IE</td>
<td>036</td>
<td>RS (record separator)</td>
<td>62</td>
<td>0x3E</td>
<td>&gt;</td>
<td>94</td>
<td>0x3E</td>
<td>&gt;</td>
</tr>
<tr>
<td>31</td>
<td>IF</td>
<td>037</td>
<td>US (unit separator)</td>
<td>63</td>
<td>0x3F</td>
<td>?</td>
<td>95</td>
<td>0x3F</td>
<td>?</td>
</tr>
</tbody>
</table>

Source: www.LookupTables.com
## Basic units of data

### Multi-byte units:

<table>
<thead>
<tr>
<th>unit</th>
<th>abbreviation</th>
<th>total bytes</th>
<th>nearest decimal equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobyte</td>
<td>KB</td>
<td>$1,024^1$</td>
<td>$1000^1$</td>
</tr>
<tr>
<td>megabyte</td>
<td>MB</td>
<td>$1,024^2$</td>
<td>$1000^2$</td>
</tr>
<tr>
<td>gigabyte</td>
<td>GB</td>
<td>$1,024^3$</td>
<td>$1000^3$</td>
</tr>
<tr>
<td>terabyte</td>
<td>TB</td>
<td>$1,024^4$</td>
<td>$1000^4$</td>
</tr>
<tr>
<td>petabyte</td>
<td>PB</td>
<td>$1,024^5$</td>
<td>$1000^5$</td>
</tr>
<tr>
<td>exabyte</td>
<td>EB</td>
<td>$1,024^6$</td>
<td>$1000^6$</td>
</tr>
<tr>
<td>zettabyte</td>
<td>ZB</td>
<td>$1,024^7$</td>
<td>$1000^7$</td>
</tr>
<tr>
<td>yottabyte</td>
<td>YB</td>
<td>$1,024^8$</td>
<td>$1000^8$</td>
</tr>
</tbody>
</table>
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- Markdown in brief
- git and Github for version control
Why we’re using R

▶ Becoming *lingua franca* of statistical analysis in academia
▶ Often demanded by employers in the private sector
▶ It’s free and open-source
▶ Flexible and extensible through *packages* (over 10,000 and counting!)
▶ Powerful tool to conduct automated text analysis, social network analysis, and data visualization, with packages such as quanteda, igraph or ggplot2
▶ Command-line interface and scripts favor reproducibility
▶ Excellent documentation and online help resources

R is also a full programming language; once you understand how to use it, you can learn other languages too.
```r
library(ggplot2)
source("plots/FormatPlot.R")
View(diamonds)
summary(diamonds)
summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price, data=diamonds, color=clarity, 
  xlab="Carat", ylab="Price", 
  main="Diamond Pricing")
format.plot(p, size=24)
```
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- R markdown in brief
- git and Github for version control
Exemplary markdown files

01-RMarkdown.Rmd
02-intro-to-R.Rmd
Plan for today

- Administration and logistics
- On the history of data and databases
- Data types and storage units
- Introduction to R
- Markdown in brief
- **git and Github for version control**
Introduction to git/GitHub

Git is a type of version control system or VCS

- A VCS keeps records of your changes: It helps track who made changes when
- Possibility of reverting any changes and go back to previous state
- Distributed (entire code and history on each machine) – allows for collaborative development
- Git: Created by Linus Torvalds in 2005 to facilitate Linux kernel development
- Other options: Mercurial, Subversion
- GitHub allows you to host repositories and adds extra functionalities (UI, documentation, issues, user profiles...)
Basic concepts of git

- Code lives in a repository: Collection of all files (and history)
- Every time you make changes, you need to make a commit:
  - Creates a snapshot of your code
  - Informs how files have changed
  - You need to add a message explaining changes
- After you commit, you need to push the changes to the repository on GitHub so that others can see them
- Note – you also need to pull first to receive changes from other people
- When you start from a repository someone created, you will have to first fork it (create a copy on GitHub) and then clone it (download) to your computer