Lecture II: Data Cleaning

BIO442: FALL 2020

DR. VIVIENNE FOROUGHIRAD

In the News

BBC NEWS

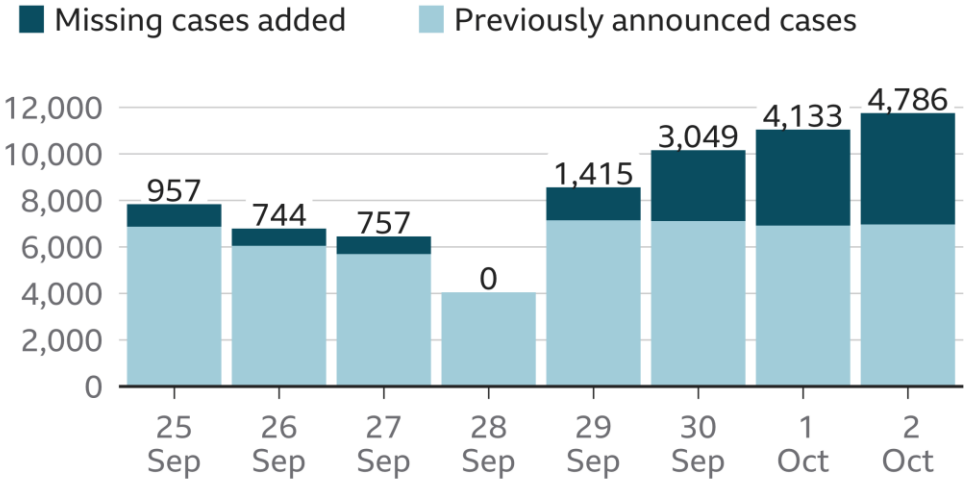
Home | Coronavirus | UK | World | Business | Politics | Tech | More

Covid: 16,000 coronavirus cases missed in daily figures after IT error

05 October 2020 | UK

Thousands of missing coronavirus cases added after reporting problem

Number of new coronavirus cases by date reported



Source: Gov.uk dashboard, Public Health England

BBC

Rules for reproducible computational research

1) Keep track of how every result was produced

2) Avoid manual data manipulation steps

3) Archive versions of programs used

4) Version control

5) Record intermediate results in standardized formats

6) Preserve how "random" results were generated

7) Always store raw data (behind visualizations)

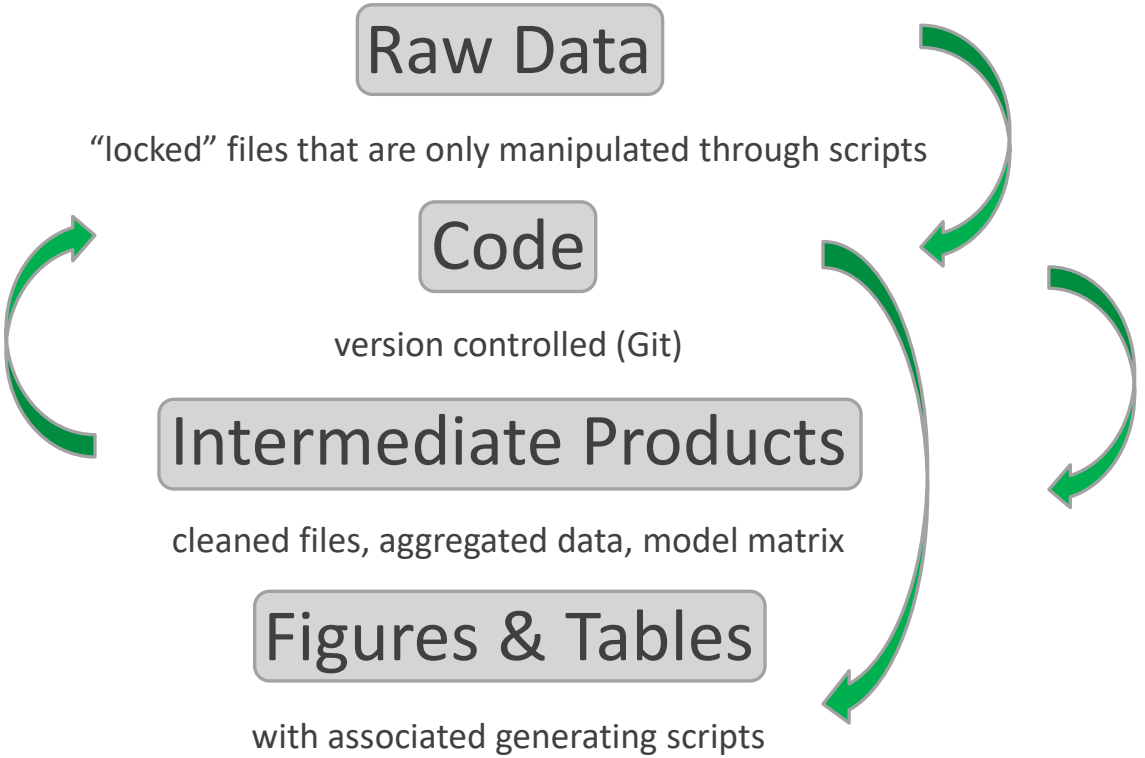
8) Hierarchical Analysis Output

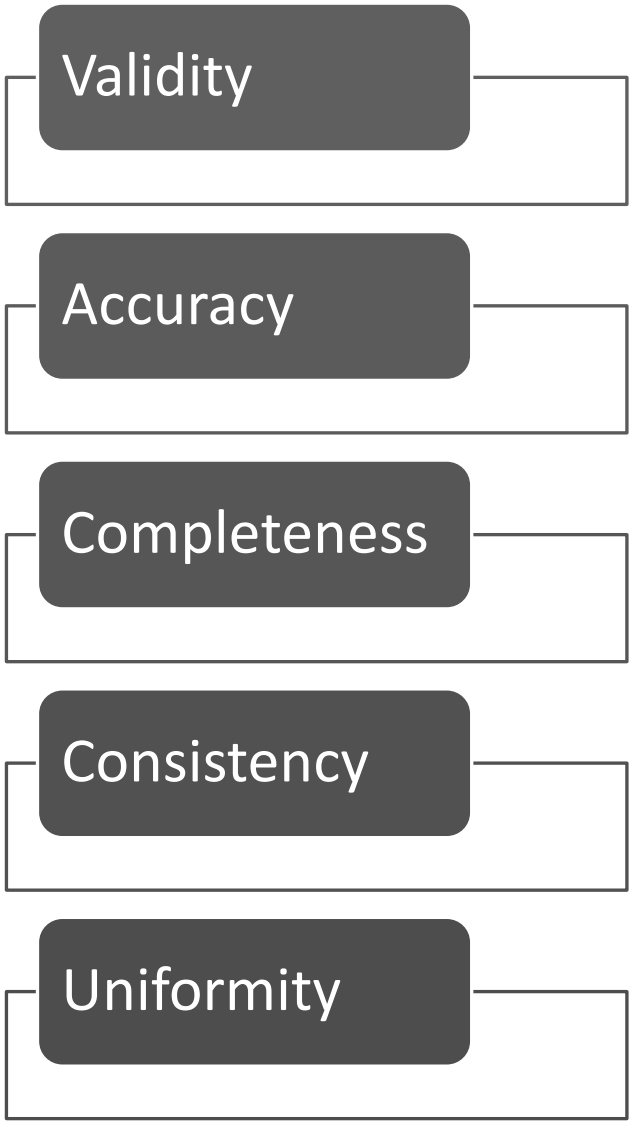
9) Connect textual statements to underlying results

10) Provide public access to scripts, runs, and results

Sandve et al. 2013

Pipeline Components





Data Cleaning

The act of cleaning data imposes values/judgments/interpretations upon data intended to allow downstream analysis algorithms to function and give results. That's exactly the same as doing data analysis. In fact, "cleaning" is just a spectrum of reusable data transformations on the path towards doing a full data analysis. -Randy Au

ALL CLEAN DATA ARE ALIKE, ALL DIRTY DATA ARE DIRTY IN THEIR OWN UNIQUE WAY

https://twitter.com/dataeditor/status/1280278987797942272

Definitions

Validity - Data are valid if it conforms to the syntax (format, type, range) of its definition

Accuracy - The degree to which data correctly describes the "real world" object or event being described

Completeness - The proportion of stored data against the potential of "100% complete"

Consistency - The absence of difference, when comparing two or more representations of a thing against a definition

Uniformity – The extent to which all sources elude to a unique value

First Name	Last Name	City	Email	Height
Vivienne	Foroughirad	Washington	vforoughirad@gmail.com	5'8"
Vivienne	Foroughriad	Washington, DC	vjf5@georgetown.edu	173cm
vivienne	Foroughirad	DC	NA	68in

Data Types

Atomic Vectors (6)

- logical
- integer
- double
- complex
- character
- raw

\+ Dimensions
Attributes

{ Factors
Date, DateTimes, Durations

{ Matrix
Dataframe
List

{ Table
Polygon
Tibble

Dates

Default date format is 4-digit year “-” month “-” day

```
Sys.Date()
```

```
## [1] "2020-10-05"
```

```
Sys.time()
```

```
## [1] "2020-10-05 14:13:59 EDT"
```

Dates can be obtained from numeric or character vectors by specifying “origin” and “format”

Character Dates

```
dat <- "13-Jan-18"

class(dat)
```

```
## [1] "character"
```

Format

```
dat <- as.Date(dat, format = "%d-%b-%y")

class(dat)
```

```
## [1] "Date"
```

Symbol	Meaning	Example
%d	day as a number (0-31)	01-31
%a %A	abbreviated weekday unabbreviated weekday	Mon Monday
%m	month (00-12)	00-12
%b %B	abbreviated month unabbreviated month	Jan January
%y %Y	2-digit year 4-digit year	07 2007

Numeric Dates

```
dat <- as.numeric(dat)
dat
```

```
## [1] 17544
```

Origin

```
#R default origin is 1-Jan-1970
as.Date(dat, origin = "1970-01-01")
```

```
## [1] "2018-01-13"
```

```
#Excel default origin is 30-Dec-1899
exceldat <- 43113
as.Date(exceldat, origin = "1899-12-30")
```

```
## [1] "2018-01-13"
```

Missing Data

Process: identify, specify, remove, retain, impute

Assign when reading in data

```
read.csv(file, na.strings = "NA")
```

Check for missing data

```
anyNA()
```

```
> x <- c(3,4,NA)
> mean(x)
```

Specify which values denote missing data

```
my_vector[my_vector=="-99"] <- NA
```

Remove missing data

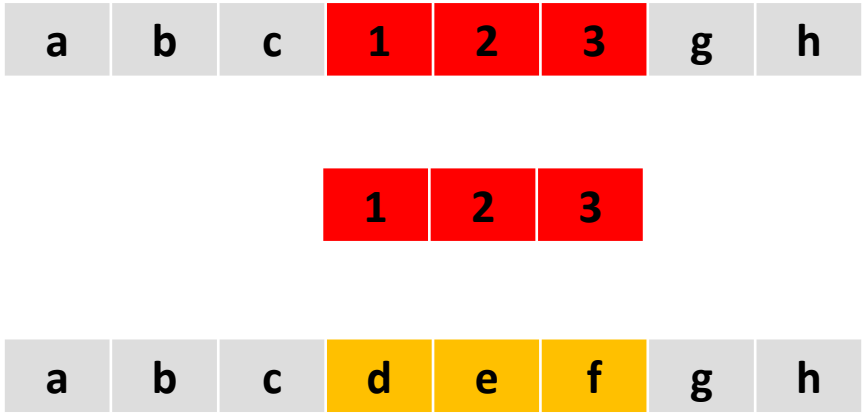
```
complete.cases(), na.omit()
```

Regular Expressions

Regular expressions are a concise and flexible tool for describing patterns in strings

```
?grep()
```

1) Detect `grep()`
2) Locate `regexpr()`
3) Extract `regmatches()`
4) Replace `gsub()`



Regular Expressions

Can match a whole string or any part of a string

```
text <- c("The","quick","brown","fox","jumps","over","the","lazy","dog")

grep(pattern = "the", x = text, ignore.case = TRUE)
```

```
## [1] 1 7
```

```
grep(pattern = "r", x = text, ignore.case = TRUE)
```

```
## [1] 3 6
```

Can match the beginning or middle of a string

Metacharacters

```
. ^ $ * + ? { } [ ] \ | ( )
```

Character Classes

Escapes

Anchors

Quantifiers

Character Classes

`.`	Any character except new line (`\n`)
`\s`	White space
`\S`	Not white space
`\d`	Digit (0-9)
`\D`	Not digit
`\w`	Word (A-Z, a-z, 0-9, or _)
`\W`	Not word

Metacharacters

ESCAPING

\'	single quote
\"	double quote
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

ANCHORS

^ or \A	Start of string
$ or \Z	End of string
\b	Word boundary
\B	Not word boundary

```
us.phones.regex <- "^\\s*(\\+\\s*1(-?|\\s+))*[0-9]{3}\\s*-?\\s*[0-9]{3}\\s*-?\\s*[0-9]{4}$"
```

Regular Expressions

Can match a whole string or any part of a string

```
text <- c("The","quick","brown","fox","jumps","over","the","lazy","dog")

grep(pattern = "the", x = text, ignore.case = TRUE)
```

```
## [1] 1 7
```

```
grep(pattern = "r", x = text, ignore.case = TRUE)
```

```
## [1] 3 6
```

Can match the beginning or middle of a string

```
grep(pattern = "^o", x = text, ignore.case = TRUE, value=TRUE)
```

```
## [1] "over"
```

```
grep(pattern = ".o.", x = text, ignore.case = TRUE, value=TRUE)
```

```
## [1] "brown" "fox"   "dog"
```

Key Cleaning Functions

`grep()` – find matching strings

`gsub()` – replace parts of strings

`strsplit()` – split a string by a delimiter

`substr()` – extract a subset of a string

`trimws()` – remove leading and trailing whitespace

`tolower(), toupper()` – convert to lower or upper case

Data Checklist

Are data appropriately parsed? Is one value recorded per cell?

`strsplit(), substr()`

Are data the right type? (numeric, character)

`class(), mode()`

Are the values plausible?

- Numeric – `min(x), max(x), hist(x)`
- Character – `length(unique(x)), table(x)`

Are data missing? Should these values be retained? Is missing data coded consistently?

`anyNA(), na.strings()`

RAW		
Name		
Jane Doe		
Height		
~2m		

PARSED		
First Name	**Last Name**	
Jane	Doe	
Certainty	**Height**	**Units**
estimate	2	meters

Database Structure

KEYS

Define unique records

Not allowed to be null

NORMAL FORMS

1NF:

A table (relation) is in *1NF* if:

1. There are no duplicated rows in the table.
2. Each cell is single-valued (no repeating groups or arrays).
3. Entries in a column (field) are of the same kind.

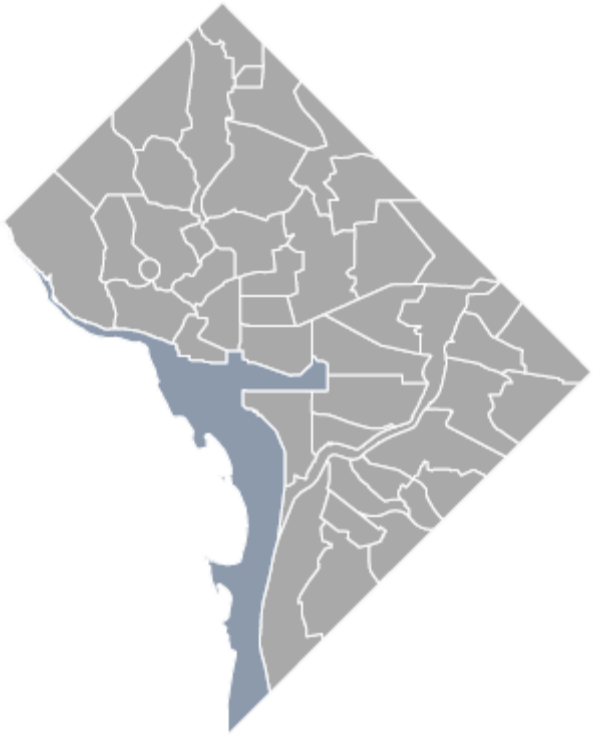
2NF:

Every non-prime attribute of the relation is dependent on the whole of every candidate key

Coding Assignment 1: Data Cleaning

Task:

After months of cajoling, you have persuaded ornithologists from three institutions in DC to agree to share their data on bird sightings as part of a collaborative project. You've now received data from these collaborators, and need to clean and combine the data and produce a map for the time period interest.



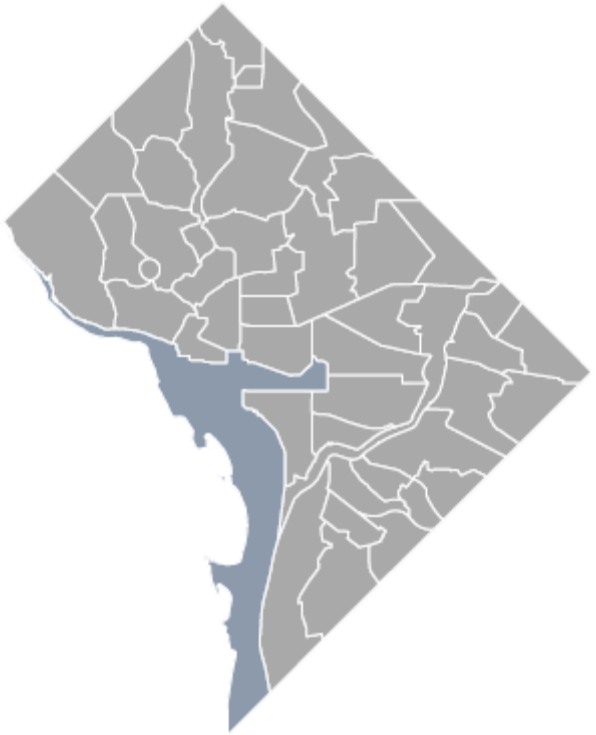
Steps:

1) Read in and combine the data
2) Standardize the data format
3) Remove and clean any erroneous values
4) Filter points to include just those taken on or after Jan, 1st, 2010
5) Filter points to include only those taken during transect surveys
6) Use the map_template.R script to plot the final set of cleaned and filtered data

Coding Assignment 1: Data Cleaning

```
head(clean_data)
```

```
##   Longitude Latitude       Date Survey_Type
## 1 -77.06518 38.90953 2010-01-01    transect
## 2 -77.01843 38.95766 2010-01-07    transect
## 3 -77.01609 38.93802 2010-01-07    transect
## 4 -77.04336 38.92275 2010-01-09    transect
## 5 -77.01796 38.94130 2010-01-15    transect
## 6 -77.03152 38.94651 2010-01-20    transect
```

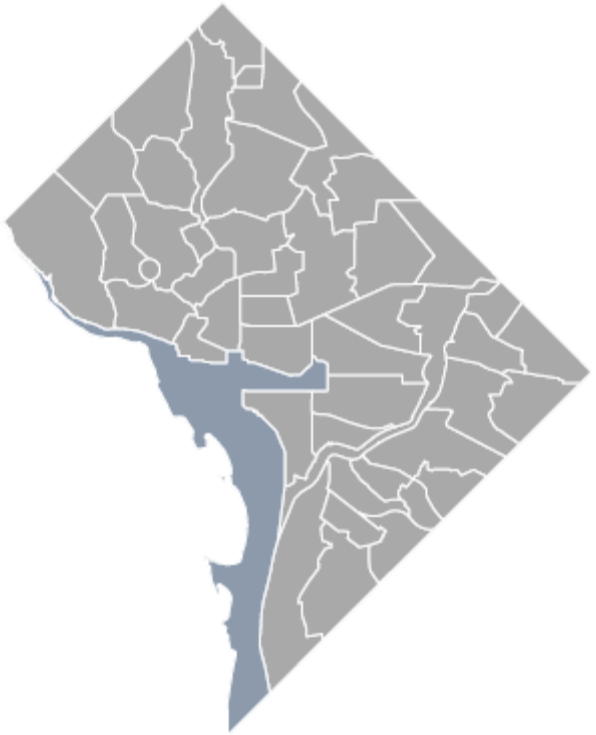


Coding Assignment 1: Data Cleaning

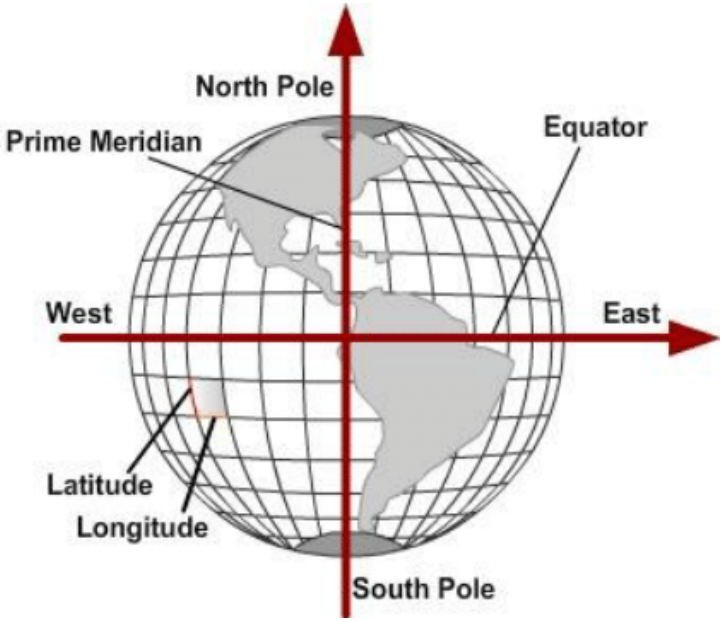
All the cleaning must be able to be reproduced using R scripts, and the original files cannot be modified. You are free to use any packages you choose, although you do not need to use any other than the ones provided for plotting the map.

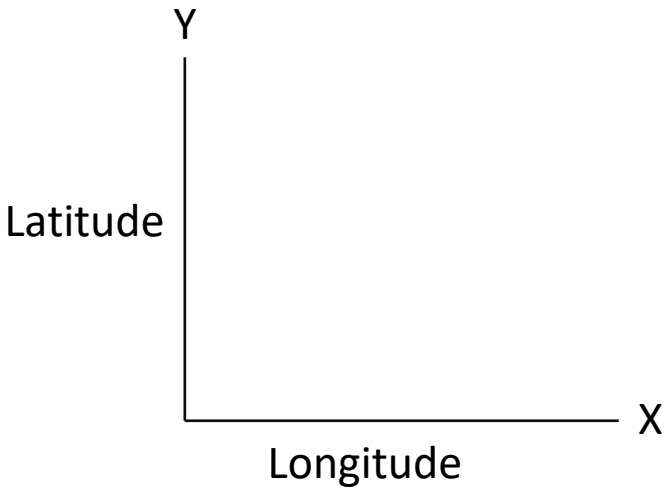
Turn in your final R script and a copy of your final map (a pdf is fine). The assignment is due October 12th, 3:59pm.

BONUS: If you complete all data cleaning using only base R you will receive 1 extra point



Spatial Data





	Latitude	Longitude
Decimal Degrees	38.91	-77.07
Decimal Minutes	38° 54.6' N	77° 4.2' W