WEEK 5

Introduction to Coding for Data Visualization

DATA VISUALIZATION

Digital storytelling at the confluence of science, art, and technology

AGENDA FOR TODAY

Uniting raster and vector graphics for our graphical abstract:

Animation Critique
Coding for DataViz

Data Visualization Animations

A critique and celebration

DATA VISUALIZATION ANIMATIONS

Critique/Celebration!

We will display a few animations and discuss each one.

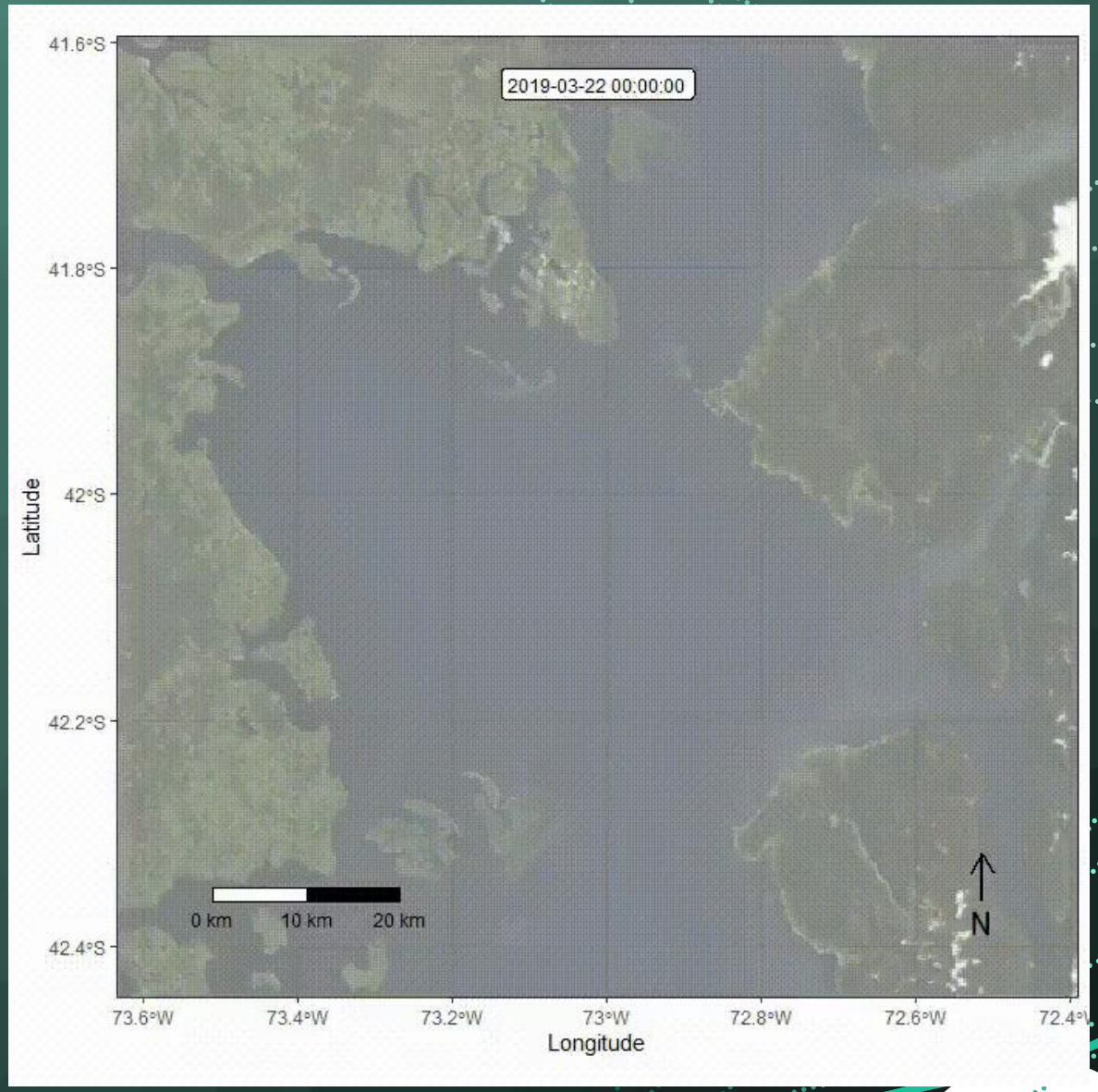
Some discussion points:

1) How do your eyes travel across the screen during the animation?
2) How do the colors, style, text, and graphics influence the message?
3) Describe the story the animation tells.
4) What questions are you still wondering after looking at the animation?

Please be mindful and conscientious while providing feedback by participating in the discussion, turning on video whenever possible, and making space for everyone to be heard. We want to create a supportive environment where we feel comfortable sharing our work.

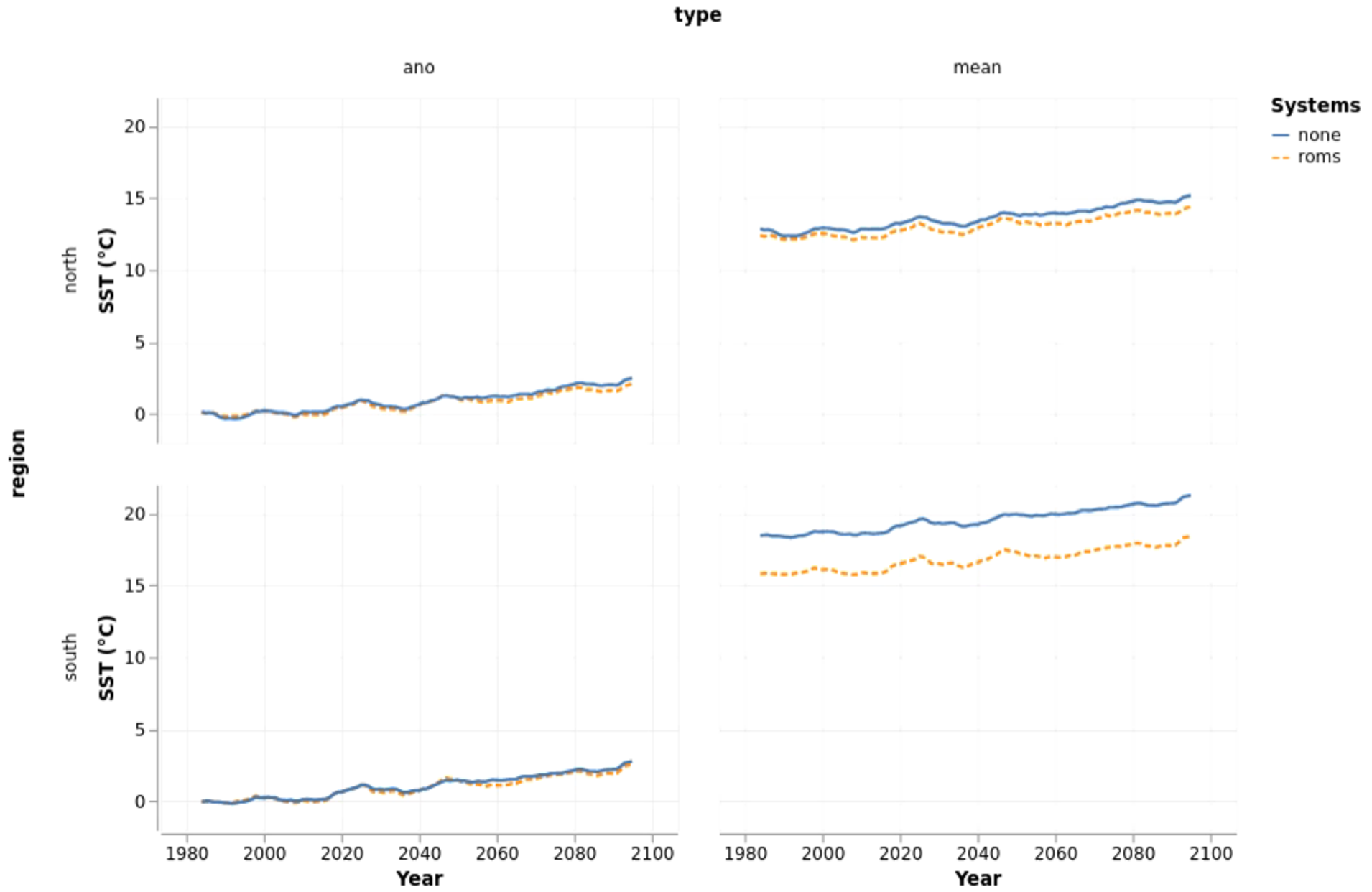
WHY ANIMATE?

Adding time to your data visualization can build anticipation, tension, and help tell your story, while making it more engaging.



This is approximately a 7 day period of Chile's Golfo de Ancud. A couple comments have indicated this article as the source study: https://www.nature.com/articles/s41598-021-82220-5

Artist: Ricardo Yarza



Artist: Ari Iramanesh – interactive web prototype using Vega Lite

Intro to Coding for Data Visualization

Intro to plotting in R, Python, and Javascript (D3).

WHAT IS CODING?

Providing a precise to-do list for your computer.

WHAT ARE CODING LANGUAGES?

Different ways of writing instructions which are translated into binary instructions given to the computer.

LOW-LEVEL VERSUS HIGH-LEVEL?

NO ABSTRACTION

Mid-Level: C

High level examples: Python, Java, C#, etc.

ABSTRACTION

Machine language (binary)

Assembly language

Programming language

LOW-LEVEL

- Closer to machine language
- Directly executable
- Very efficient- great for firmware
- Hard to read, write, and maintain

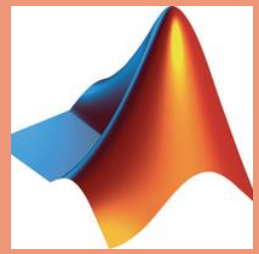
HIGH-LEVEL

- Closer to human language
- Translated via compiler/interpreter
- Programming paradigms:
  - Procedural (sequence of instructions) top down approach
  - Object-oriented (interactions between objects) bottom up approach
  - Multi-paradigm

WHAT SHOULD WE USE?

Finding your hammer.

MATLAB (not free): High-level multi-paradigm programming language and interactive environment for numerical computation, visualization, and programming



R LANGUAGE

- High-level language and environment extensively used by ecologists and scientists: stats & dataviz
- PROS: open-source, large community developing packages

JAVASCRIPT

- High-level interpreted multi-paradigm programming language – primarily for Web Pages
- PROS: fast, ubiquitous, relatively easy, many dataviz libraries

D3.js

VEGA

Vue.js

Observable

PYTHON

- High-level object-oriented programming language
- PROS: open source, fast for large datasets, dynamic typing, large community

matplotlib

seaborn

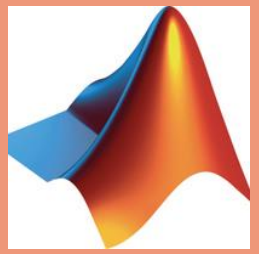
ggplot2

Jupyter Notebook

FINDING YOUR FLOW

Exploring workflows in R, Javascript, and Python.

MATLAB (not free): High-level multi-paradigm programming language and interactive environment for numerical computation, visualization, and programming



R Studio

Install packages with RSTUDIO CONSOLE

EDIT & RUN CODE:

RSTUDIO

PUBLISH & SHARE CODE

RMARKDOWN

LET PEOPLE INTERACT:

RSHINY

JS

Observable: Include libraries by:

d3=require('d3');

HTML: Include libraries inline

<script src="link"></script>

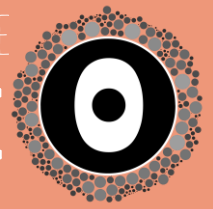
EDIT & RUN CODE:

VISUAL STUDIO CODE

with HTML, CSS, JAVASCRIPT

PUBLISH & SHARE CODE

OBSERVABLE



LET PEOPLE INTERACT:

PUBLISH WEBPAGE

Install packages through package manager like ANACONDA or in your terminal:

```
conda install pip
```

EDIT & RUN CODE:

SPYDER

through ANACONDA

PUBLISH & SHARE CODE

JUPYTER NOTEBOOK

LET PEOPLE INTERACT:

PUBLISH WEBPAGE

EXPLORING BASIC FUNCTIONS

Workflows.

R LANGUAGE

Open RSTUDIO, New Project (sets directory)

Install R Markdown:

```
install.packages("rmarkdown")
```

Make a header:

```
# Header 1
## Header 2 etc.
```

JAVASCRIPT

Open ObservableHQ

Make a title:

```
md`# Your title here`
```

PYTHON

Open Spyder

Install matplotlib, seaborn, numpy, scipy, pandas..

```
conda install matplotlib
Etc..
```

INITIALIZE A VARIABLE

R LANGUAGE

```
data <- 2
```

JAVASCRIPT

```
var data = 2;
```

PYTHON

```
data = 2
```

MAKE A COMMENT

R LANGUAGE

#I'm just talking to myself

JAVASCRIPT

//I'm just talking to myself

PYTHON

#I'm just talking to myself

FOR LOOP

Telling the computer "cycle through each one of these values (also called "interating over" each value). For example, going through each row of a dataset from top to bottom or each column from left to right. Notice that some programs (Python) need indentation to understand what you're trying to do, while others (like R and Javascript) use curly brackets for this purpose.

R LANGUAGE

```
for (x in 1:10) {
print(x)
}
```

JAVASCRIPT

```
var i;
for (i = 0; i < cars.length; i++) {
text += cars[i] + "<br>";
}
```

PYTHON

```
fruits = ["apple", "banana",
"cherry"]
for x in fruits:
print(x)
```

IF/ELSE STATEMENT

Telling the computer "If the previous conditions were not true, then try this condition". The "else if" command is what it goes to next, and then the "else" is a catch all.

R LANGUAGE

```
a <- 200
b <- 33

if (b > a) {
print("b is greater than a")
} else if (a == b) {
print("a and b are equal")
} else {
print("a is greater than b")
}
```

JAVASCRIPT

```
if (time < 10) {
greeting = "Good morning";
} else if (time < 20) {
greeting = "Good day";
} else {
greeting = "Good evening";
}
```

PYTHON

```
a = 200
b = 33
if b > a:
print("b is greater than a")
elif a == b:
print("a and b are equal")
else:
print("a is greater than b")
```

PRINTING AN OUTPUT

R LANGUAGE

R's default is to print an output, but you may need this if you want to print something within a for loop.

print()

and paste()

JAVASCRIPT

console.log("Hello World");

Javascript does NOT print outputs by default.

PYTHON

print()

Python does NOT print outputs by default.

POST-MEETING TO DO'S

- Due next week: Publication-ready Figure
  - Use one of these three programming languages to code your data visualization figure. Include your code in your submission.
  - For R/Rmarkdown, knit to PDF or html, for Javascript, share an Observable notebook or a folder with HTML, CSS, and Javascript, and for Python, share a .py script or a Jupyter notebook.
- Next week, we will be exploring web design and data visualization. I will send out a series of tutorials which will lay down the basics of web design using HTML, CSS, and Javascript together to create a custom webpage.