Data-Driven Animation

for Science Communication

Jessica Kendall-Bar

July 18-29
CSUMB Summer of Science Illustration

OUTLINE

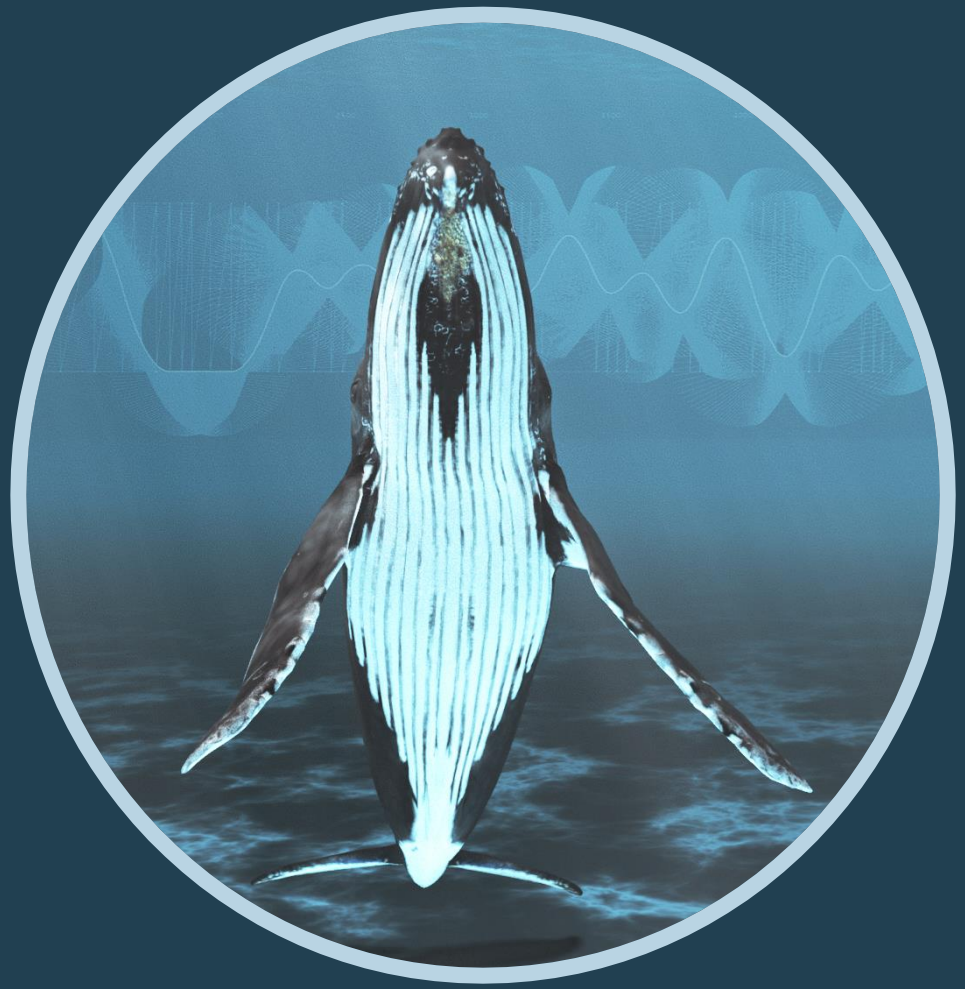
Course structure

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
	Jul 18	Jul 19	Jul 20	Jul 21	Jul 22	Jul 25	Jul 26	Jul 27	Jul 28	Jul 29
Topic	***Introductions & Collaboration***	***Conceptualization***	***Digital Art and Design***	***Data Visualization***	***Animation & Motion Graphics***	***Animating 2D Data***	***Intro to 3D Animation***	***Animating 3D Data***	***Editing & Compositing***	***Final Critique & Communication Strategy***
Program	Adobe Premiere Pro	Adobe Premiere Pro	Adobe Photoshop & Illustrator	Adobe Illustrator & R	After Effects	After Effects	Autodesk Maya	Autodesk Maya	Adobe After Effects/Premiere Pro	Social Media
Starting products	Draft scientific paper with key figures and points, any existing assets (videos, illustrations, or 3D models).	Style Guide & Script	Reference images and list of key assets and animatic	One key figure	Animatic	Made-over key figure	3D asset	Animated 3D scene	Rendered image sequences	Final composition
Main output	Style Guide, Budget, Contract, & Script	Storyboard & Animatic	Custom 2D raster & vector graphics	Made-over key figure	Scene with motion graphics	Animated key figure	Animated 3D scene	Data-driven 3D scene	Final composition (mp4 with narration, annotations, custom musical score)	Science Communication Strategy document
Interview with industry Expert - Guest lecture	**Sara El Shafie** - Distilling Complexity; Science through Story	**Ilgavak (Peter Williams)** - Storytelling and place-based learning	**Alex Boersma** - Scientific illustrations and animations for high-impact science	**Allison Horst** - Art to build agency for budding data scientists	**Bailee Desrocher** - Animations for science storytelling: NASA & beyond	**Katherine Vlamis** – CrossTown LA COVID data visualizations for large cities amidst a global pandemic	**Johnson Martin** - Digital Life 3D: Realistic animal models for science and education	**Geraldine Fauville;** VR for environmental education	**Connor Vance** - Musical Compositions for Data Storytelling	**Sal Alper** - Science communication and outreach through the SF Exploratorium

DAY 4

Data Visualization

Data-Driven Animation
for Science Communication



PART 1

WELCOME

Agenda & Goals

Review today's agenda & goals – 20 min

AGENDA

Today's agenda

2:00 - 2:20: Decoding Code – Run & comment your partner's code (20 min)
2:20 - 2:50: Code Reflection – Discuss the code (30 min)
2:50 - 3:00: Break / Reflection (10 min)
3:00 - 3:30: Lecture: Coding & Data Visualization (30 min)
3:30 - 4:00: Partner Work Time - Discuss figure improvements (30 min)
4:00 - 4:10: Break (10 min)
4:10 - 4:40: Demo: Coding in R – Editing figures with ggplot in R (30 min)
4:40 - 5:20: Work time – Edit key figures (40 min)
5:20 - 5:30: Break (10 min)
5:30 - 6:30: Guest speaker – Allison Horst: Art to build agency for budding data scientists

DAY 4

GUEST SPEAKER

Art to build agency for budding data scientists

Allison Horst

Assistant Teaching Professor: UC Santa Barbara

Contact: ahorst@ucsb.edu

GOALS

Today's goals

- Run and refine code, assess key figure
- Improve key figure

PART 2

OPENING ACTIVITY

Decoding Code & Code Reflections

Run, comment, discuss, reflect - 50 min

Opening Activity

Decoding Code

- Run & comment your partner's code

Opening Activity

Code Reflection

- Discuss the code

BREAK

10 min - Return at 3:00 PM

PART 3

LECTURE

Coding & Data Visualizations

Data Visualization Makeover - 30 min

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.

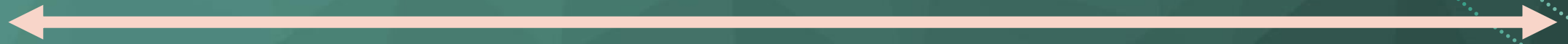
LOW-LEVEL VERSUS HIGH-LEVEL?

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

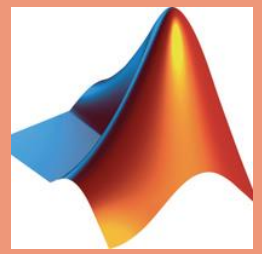
HIGH-LEVEL

- 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

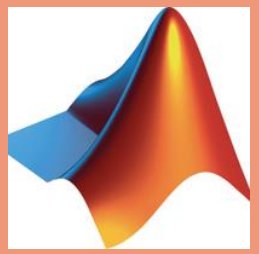
ggplot2

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>

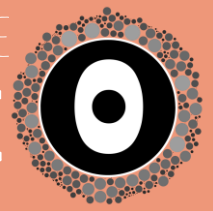
EDIT & RUN CODE:

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");

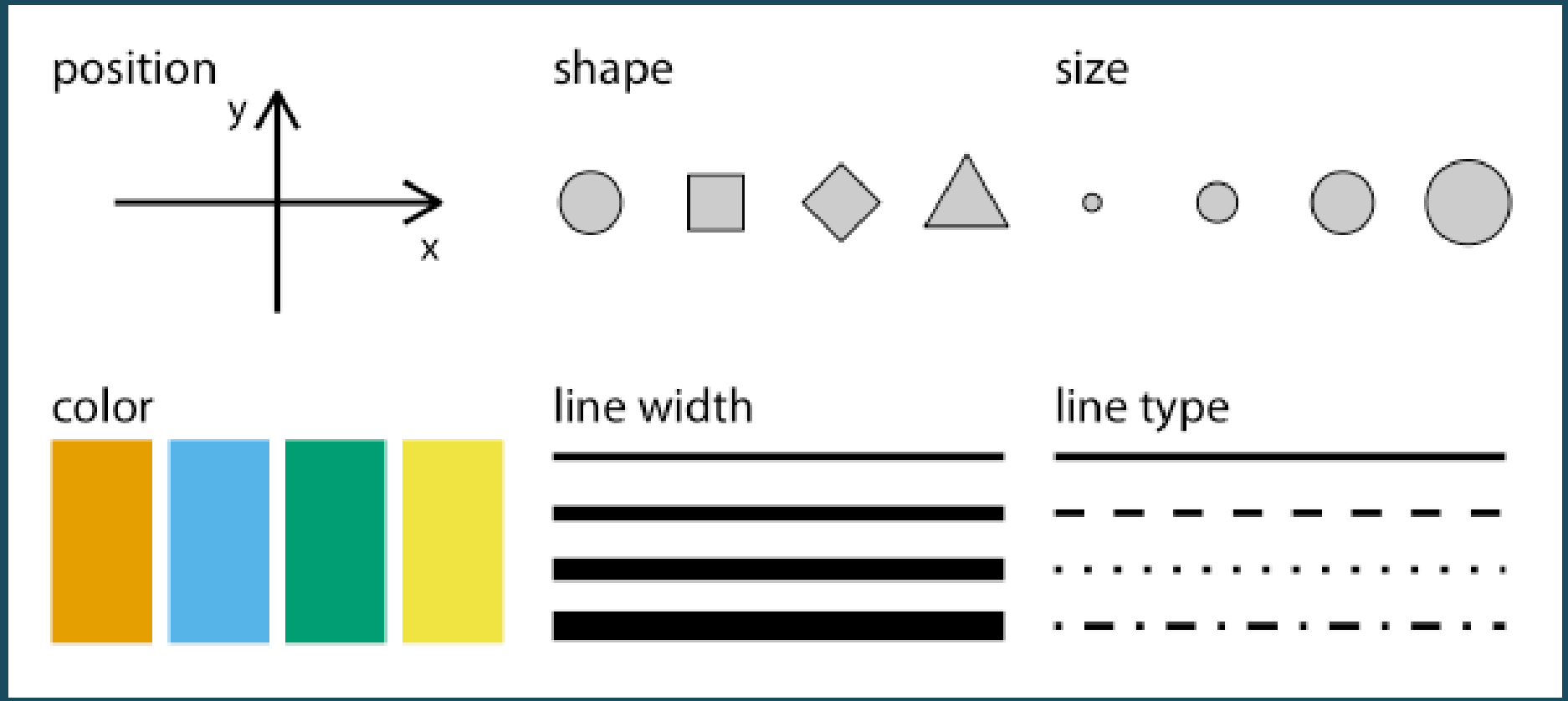
Javascript does NOT print outputs by default.

PYTHON

print()

Python does NOT print outputs by default.

MAPPING DATA ONTO SCALES



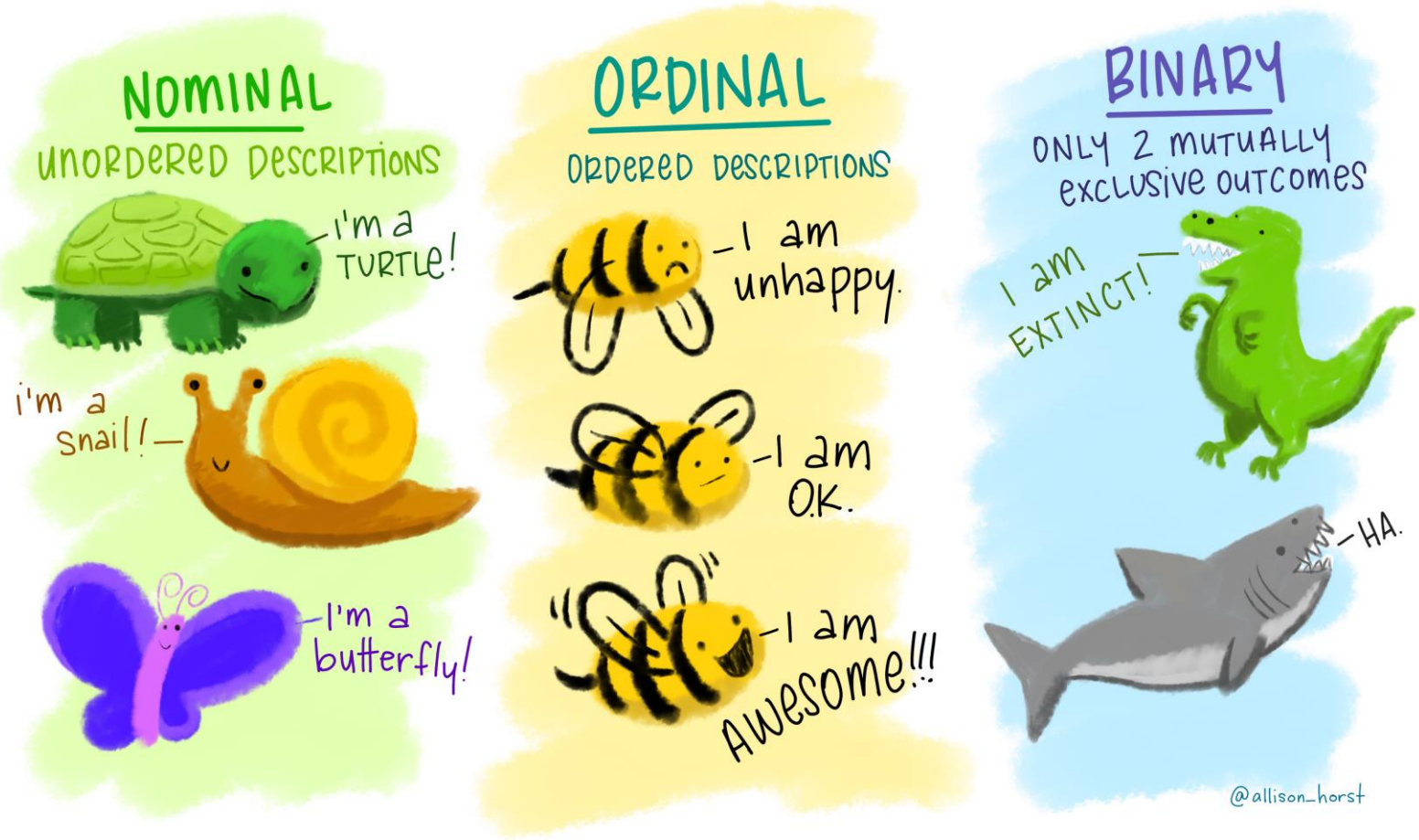
Data Visualization by Claus Wilke

+ others: 3D, time

TYPES OF DATA Quantitative / Numeric

Artwork by @allison_horst

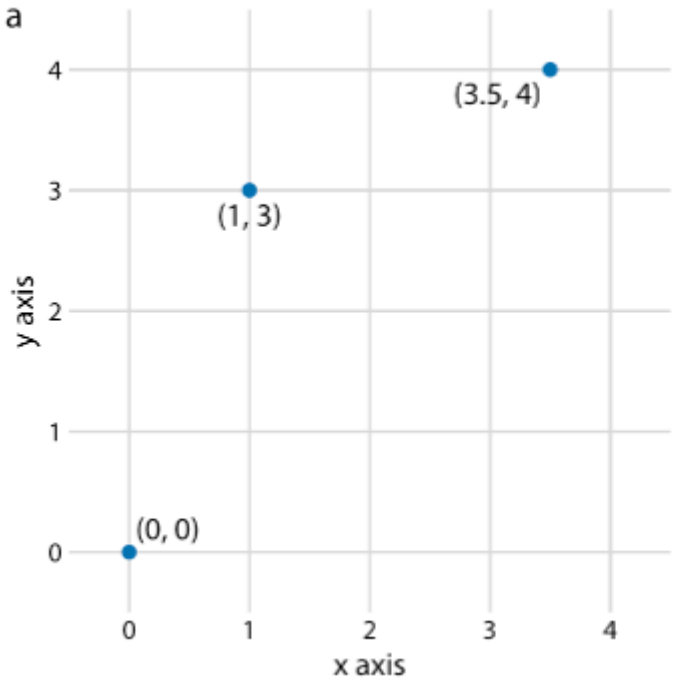
TYPES OF DATA Qualitative / Categorical



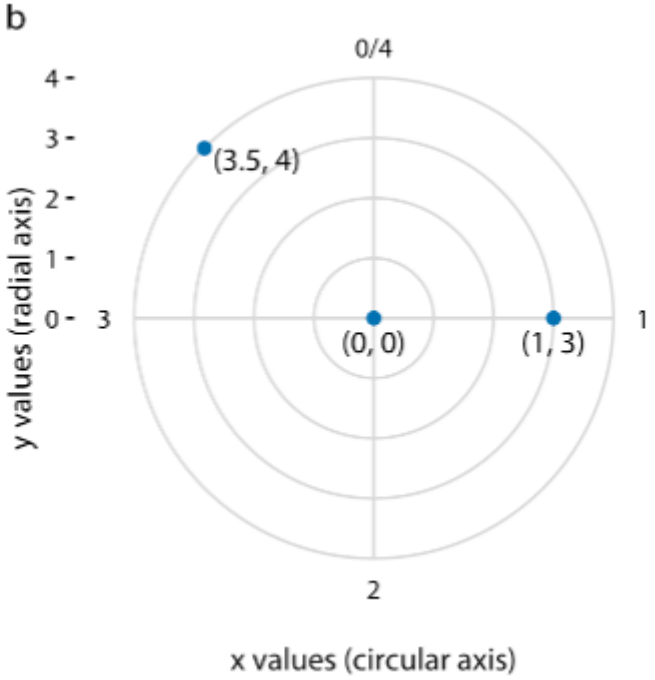
Artwork by @allison_horst

TYPES OF POSITION

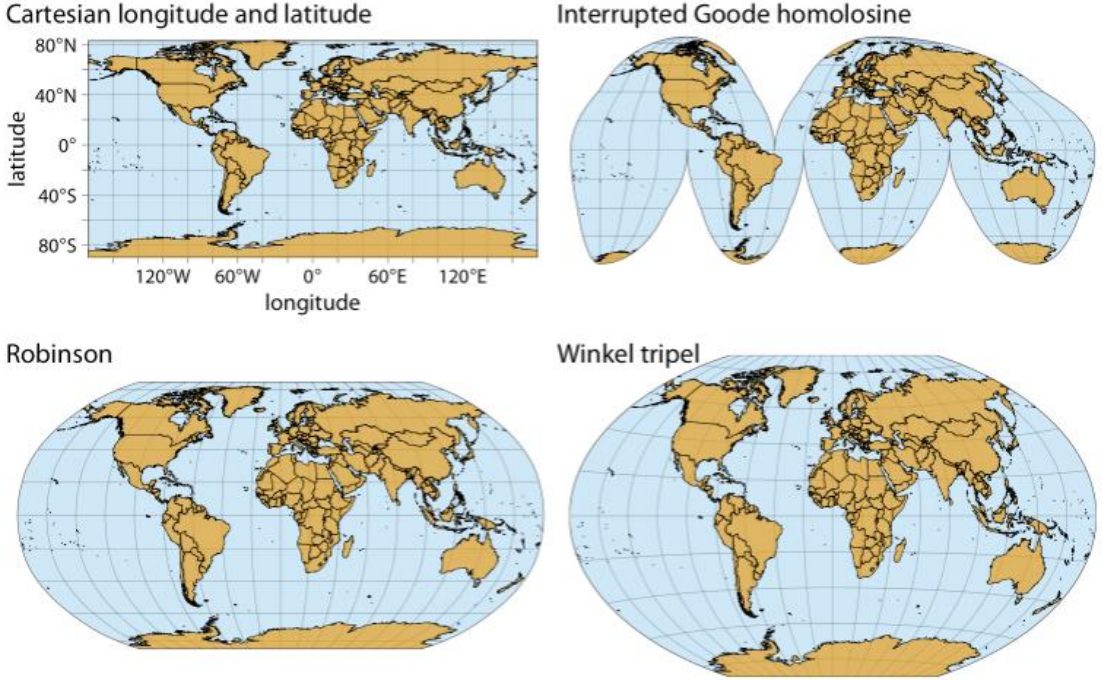
Cartesian



Polar



Geospatial data



GENERAL GUIDELINES

Hierarchy of data visualization:

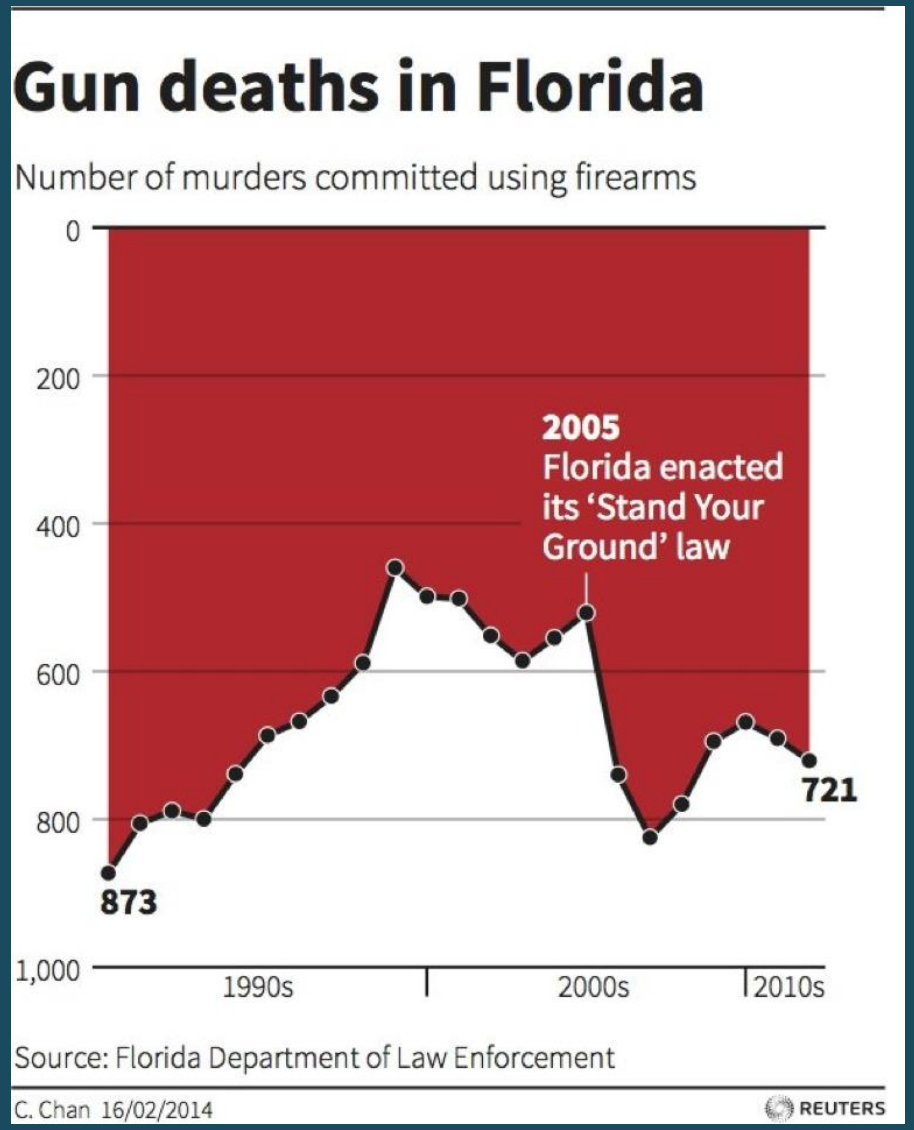
1) Is it *correct*?
2) Is it *clear*?
3) Does it communicate the data *responsibly*?
4) Does it look *awesome*?

Adapted from Allison Horst's Data Visualization lecture

EXERCISE

- Read the data visualization.

  Explain where your eye is drawn first and what the data says.



Adapted from Allison Horst's Data Visualization lecture

1) IS IT CORRECT?

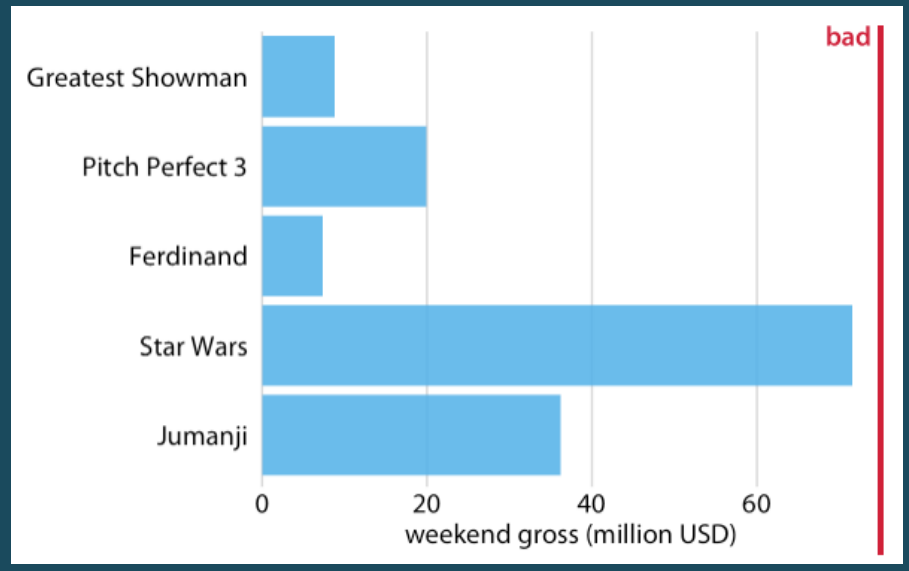
- Experimental design
- Data wrangling
  - Keep and display raw data whenever possible
  - Triple check your data wrangling steps
  - Annotate your code
  - Have someone check your code
  - Check conclusions
- Keep your axes accurate!

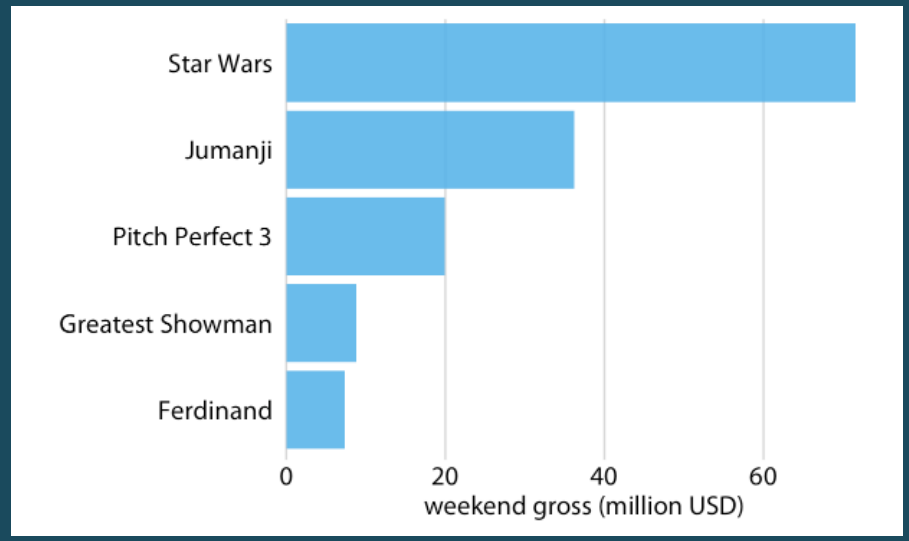
Adapted from Allison Horst's Data Visualization lecture

Data Visualization by Claus Wilke

2) IS IT CLEAR?

- Everything present should serve a purpose
- Simplify your message and story
- Emphasize your takeaway
- Order your variables (largest to smallest)
- Avoid rotated labels
- Legends: minimize use, simplify, and consider placement

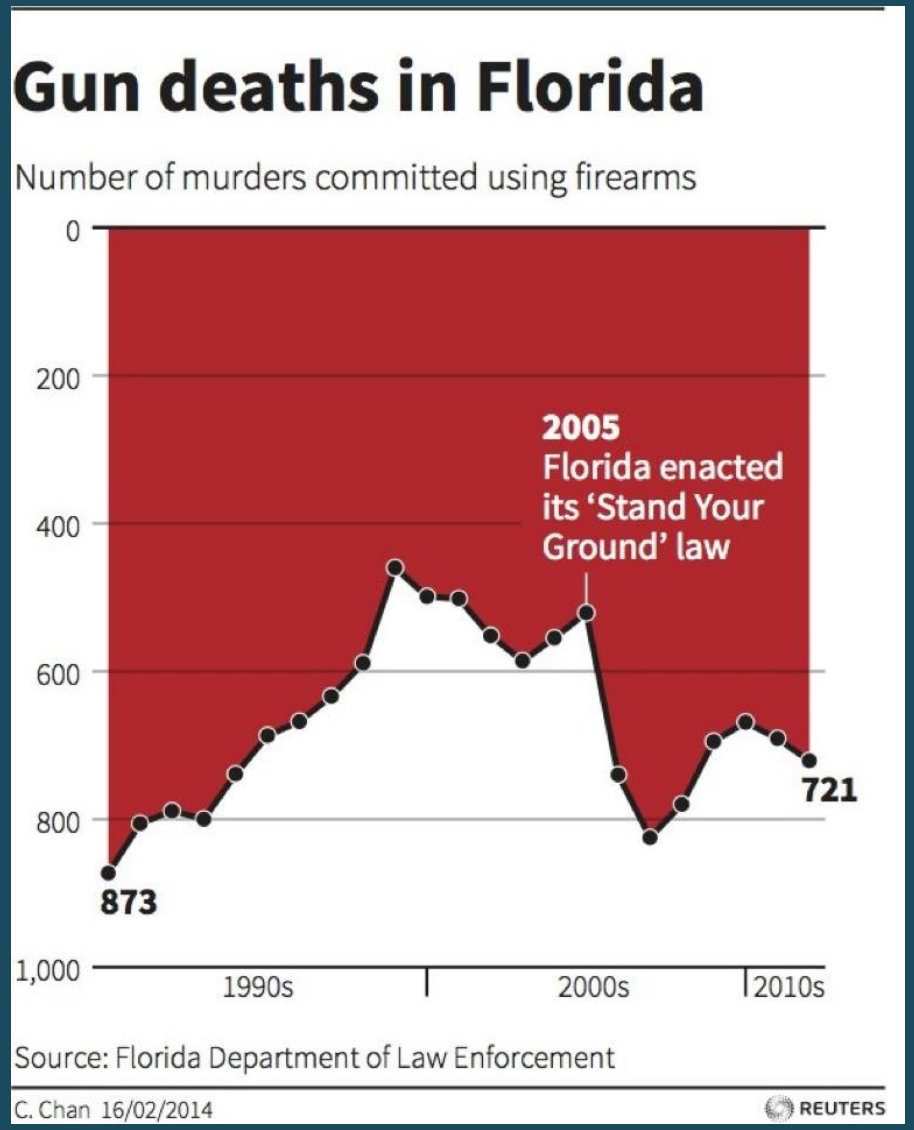




Adapted from Allison Horst's Data Visualization lecture

3) IS IT RESPONSIBLE?

- Although they may be correct, some data visualizations are misleading



Adapted from Allison Horst's Data Visualization lecture

4) DOES IT LOOK AWESOME?

- …transition to digital art & design!

back to

Illustrator

Basics of editing figures & compositing

WHY EDIT A FIGURE

GUIDELINES

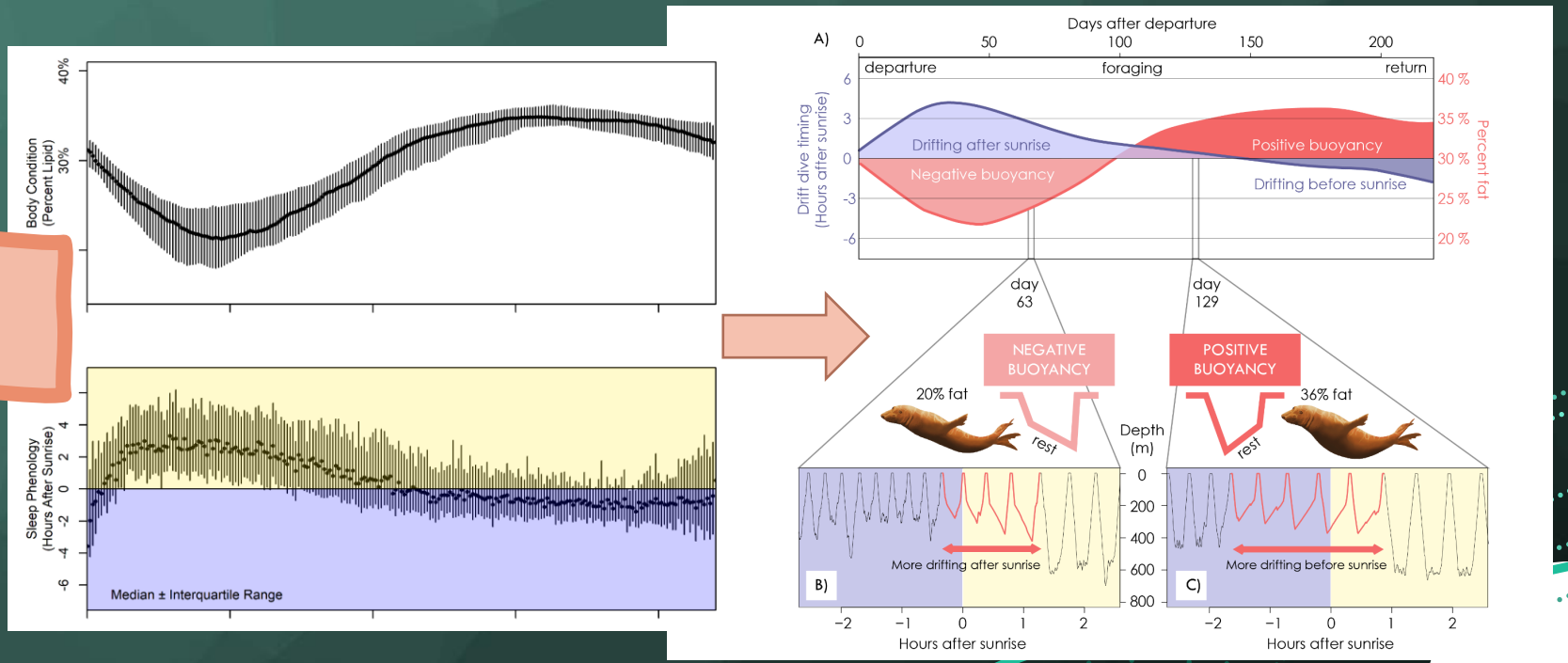
You might need to edit the font, colors, or formatting to fit the guidelines of a journal.

COMPOSITION

You may want to combine your figure with other graphics which were made separately.

AESTHETICS

You might want quick access to play with color palettes, font styles, and appearance.



WHEN TO EDIT A FIGURE IN ILLUSTRATOR

YES, DO IT

- You won't be changing the analysis or data
- You will only be making small changes to the data, but the axes will not change
- You have a reproducible workflow for generating the figure up to this point that will look like the one edited in Illustrator
- You are seeking to make small cosmetic changes such as adjustments to:
  - Marker fill color
  - Line type or stroke
  - Font (size and type)
  - Formatting scale bars and labels
  - Arranging subpanels

NO, BETTER NOT..

- You will be making changes to the data and analyses that will change the axes
- You are seeking to make extensive changes including changes like:
  - Changing axes
  - Plotting additional data
  - Marker shape
  - Combining datasets

EDITING FIGURES IN ILLUSTRATOR

KEY TOOLS:

1. Selection tool (and Direct Selection)
2. Layers/Paste in Place
3. Clipping Masks
4. "Transform Each"
5. Groups
6. Eyedropper tool

KEY STEPS:

1. Export figure as .pdf or other vector graphic format.
2. Open figure in Illustrator
3. Release clipping mask
4. Group elements
5. "Transform each" for points
6. Eyedropper tool for line styles
7. Export figure as .png

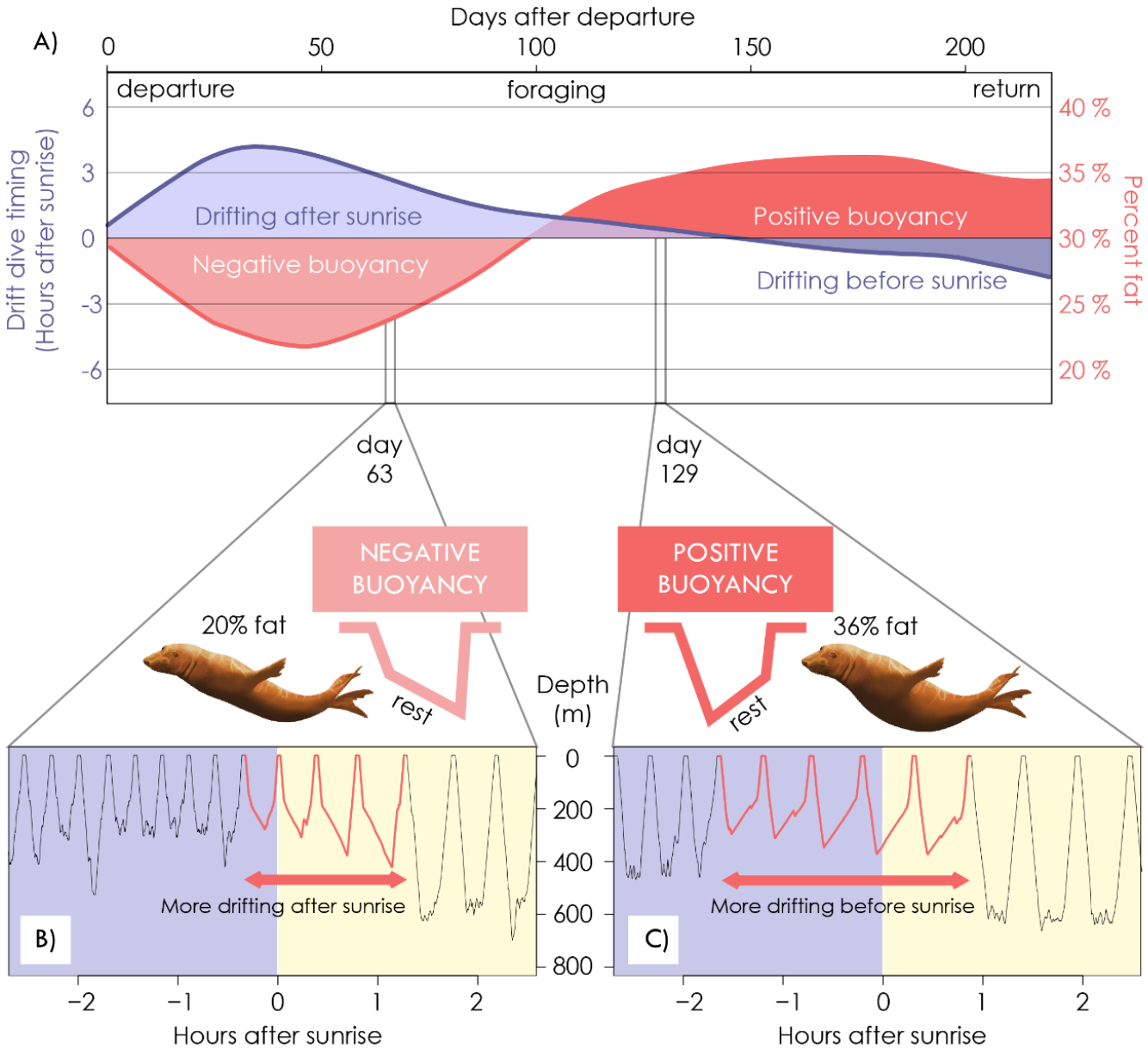
COMPOSITING IN ILLUSTRATOR

KEY TOOLS:

1. Selection tool (and Direct Selection)
2. Layers/Paste in Place
3. Clipping Masks - Create
4. “Transform Each”

KEY STEPS:

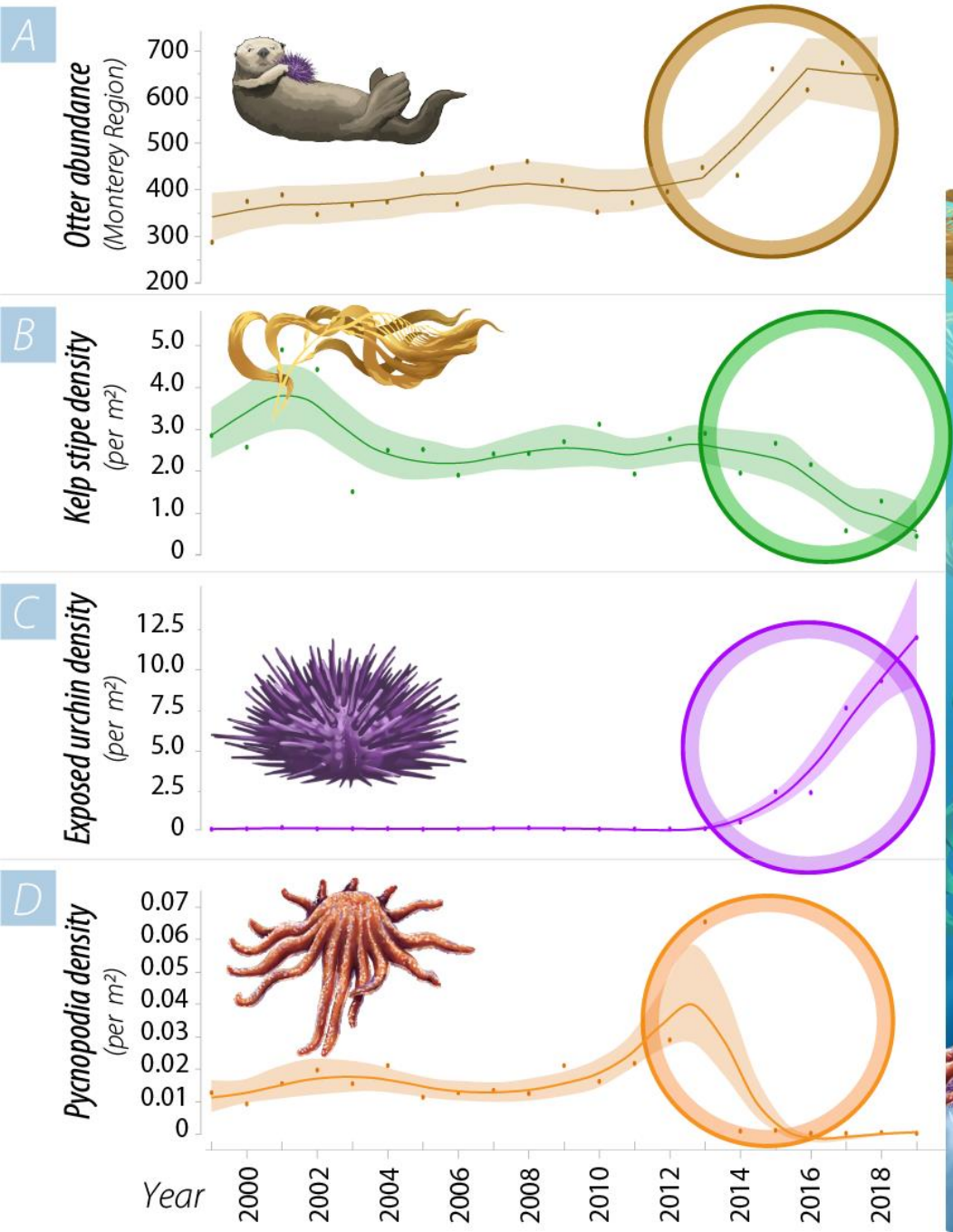
1. Insert your graphics from CC Libraries (other sources too: nounproject etc.)
2. Decide on a layout which enforces the narrative.
3. Arrange graphics
4. Make clipping masks to frame your graphics



Beltran, Kendall-Bar et al. Lightscapes of fear: how mesopredators balance starvation and predation in the open ocean. In Press. Science Advances

Demonstrating changes across time

Figures made in R, edited in Illustrator.
Seal raster graphics painted in Photoshop.


A
Otter abundance
(Monterey Region)
700
600
500
400
300
200
B
Kelp stipe density
(per m²)
5.0
4.0
3.0
2.0
1.0
0
C
Exposed urchin density
(per m²)
12.5
10.0
7.5
5.0
2.5
0
D
Pycnopodia density
(per m²)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Year
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018


Across the mosaic, sea otter preference for healthy urchins indirectly enhances forest patch resistance to overgrazing.
Active foraging urchins in barrens are starved, but passive foraging urchins in forests are healthy.
Released from predation pressure, purple sea urchins shift their behavior and transform the landscape to a patchy mosaic of barrens and remnant forests.
Sea star wasting syndrome locally extirpates the urchin predator, Pycnopodia helianthoides.
© Jessica Kendall-Bar 2020

PART 4

WORK TIME

30 min – Stop & break at 4:00 PM

BREAK

10 min - Return at 3:00 PM

PART 5

DEMO

Coding in R

Editing figures with ggplot in R - 30 min

PART 6

WORK TIME

40 min – Stop & break at 5:20 PM

BREAK

10 min - Return at 5:30 PM

HOMEWORK

Deliverables for Tomorrow

DAY 4

GUEST SPEAKER

Art to build agency for budding data scientists

Allison Horst

Assistant Teaching Professor: UC Santa Barbara

Contact: ahorst@ucsb.edu