THE LABYRINTH

DAILY NEXUS

Data Journalism Guidelines

Daily Nexus: The Labyrinth

Stella Jia Siddharth Chattoraj

January 15, 2023

Contents

1 The Process

1.1 Timeline of Producing an Article

1. Think of a project idea. Find a group if you would like to work together with another data reporter.

2. Brainstorm **storylines** for that project topic. The story is the most important part of the article.

3. Collect data, whether it be through PRA Requests (requesting data email template), Google searching, web scraping, e-mail requests, downloading/formatting pre-existing data, or other methods. You almost always need the data to be in a .csv file for best results. This may mean manually transcribing data, if it is in pdf form, into an Excel sheet and then saving the file as .csv.

4. Chart or graphic design the data in a visually appealing way that conveys a main message. Any software is allowed!

   - Examples of Coding Tools With Documentation: Python's Plotly, MatPlotLib, Seaborn; R's ggplot; JavaScript's D3.js
   - Highly recommend using AI tools like ChatGPT if you are new to coding. You can ask the program questions like "How to create a bar plot in Python MatPlotLib" and it will output code you can use.
   - Examples of Non-Coding Tools: Datawrapper, Figma, Flourish, Canva, Adobe InDesign, Excel

5. Share the graphs with the data editors and make necessary changes. The reason graphs are shared before writing the article is so that the data editors can check if the graphs accurately, following statistical principles, convey the message implied. Feel free to ask the data editors for help if you get stuck!

6. Interview sources. Reach out to people to interview who are part of or represent each side/perspective that the article's storyline may touch on. Interviews are required for most articles.

7. Write the article. Focus on conveying the story, and ensure there is an analysis that intersects background information, the quotes, and the data.

1.2 Timeline of Publishing an Article

1. Following the timeline above, writers turn in their finalized article and chart captions latest by Friday night of the week before it will be published to the data editors, who will revise them and be in communication for further editing

2. Writers fix all edits by Sunday night. Data editors will share the article and chart captions with the editor-in-chief (for print articles) and copy team that night for additional editing.

3. Writers fix all additional edits by Tuesday at 5 PM. Data editors will submit the finished work, if it will go in the weekly print edition, to the production team.

4. The article is published in print and/or online on Thursday. You will get paid 12 dollars for contributing to a write-up or design, for each set of replicable graphs (where the code is mostly the same but the data changes), and for each distinct chart you make. There is no trial period — you get paid from the very first article!

2 Brainstorming Ideas

2.1 Finding Ideas

The best way to brainstorm is not sitting down and waiting but collecting ideas over time. There's a reason why some say their best ideas come while showering. You need to give your mind space to think freely to generate good ideas. I highly recommend keeping a note on your phone or somewhere easily accessible to capture any potential ideas even if small. It often reaps greater rewards than forcing yourself to brainstorm.

2.2 Exploring Ideas

All ideas are great, but in our case we also need to think about the data journalistic value of the idea. Please think about your answer to the following questions before proceeding with the project:

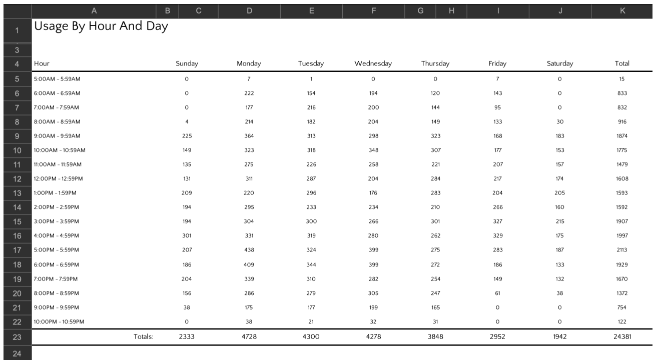
- Where could we collect data for this topic?
- How does data bring greater meaning/insight to this topic?
- Who could we interview to bring greater meaning/insight to the data?

3 Collecting and Cleaning Data

3.1 Existing Data

There is a plethora of existing data on the internet. However, it is crucial that the data found is reliable and relevant to your project. Some examples of reliable data sources include those provided by official organizations: government, university, NFL, etc. Depending on the domain of your project, pinpoint the organization involved with that type of data and search for it online. A common source of data we've used in the past is requesting it from the university, however be prepared for a lengthy wait time.

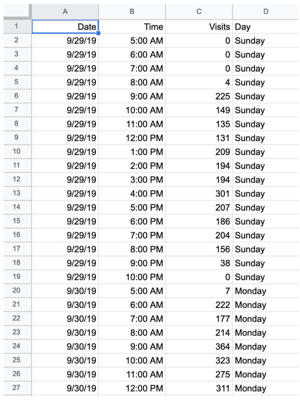
Ideally, the data should be found in a **.csv or .xlsx file** type. This allows programming languages to easily take it in. Sometimes, the data may be in an ideal file type but the file itself is organized in a manner that would be difficult for a computer to take in. In the following example, the initial data is given in a .csv file, however, there are bold lines separating the categories in the file and unnecessary white space. The bold lines will disturb the process of taking in the data when inputting it into Python or R. The organized .csv file on the right removes the unnecessary elements and maintains a default formatting style.



Usage By Hour And Day

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Total
5:00AM - 5:59AM	0	7	1	0	0	7	0	15
6:00AM - 6:59AM	0	222	154	194	120	143	0	833
7:00AM - 7:59AM	0	177	216	200	144	95	0	832
8:00AM - 8:59AM	4	214	182	204	149	133	30	916
9:00AM - 9:59AM	225	364	313	298	323	168	183	1874
10:00AM - 10:59AM	149	323	318	348	307	177	153	1775
11:00AM - 11:59AM	135	275	226	258	221	207	157	1479
12:00PM - 12:59PM	131	311	287	204	284	217	174	1608
1:00PM - 1:59PM	209	220	296	176	283	204	205	1593
2:00PM - 2:59PM	194	295	233	234	210	266	160	1592
3:00PM - 3:59PM	194	304	300	266	301	327	215	1907
4:00PM - 4:59PM	301	331	319	280	262	329	175	1997
5:00PM - 5:59PM	207	438	324	399	275	283	187	2113
6:00PM - 6:59PM	186	409	344	399	272	186	133	1929
7:00PM - 7:59PM	204	339	310	282	254	149	132	1670
8:00PM - 8:59PM	156	286	279	305	247	61	38	1372
9:00PM - 9:59PM	38	175	177	199	165	0	0	754
10:00PM - 10:59PM	0	38	21	32	31	0	0	122
Totals:	2333	4728	4300	4278	3848	2952	1942	24381

Figure 1: Initial data



Date	Time	Visits	Day
9/29/19	5:00 AM	0	Sunday
9/29/19	6:00 AM	0	Sunday
9/29/19	7:00 AM	0	Sunday
9/29/19	8:00 AM	4	Sunday
9/29/19	9:00 AM	225	Sunday
9/29/19	10:00 AM	149	Sunday
9/29/19	11:00 AM	135	Sunday
9/29/19	12:00 PM	131	Sunday
9/29/19	1:00 PM	209	Sunday
9/29/19	2:00 PM	194	Sunday
9/29/19	3:00 PM	194	Sunday
9/29/19	4:00 PM	301	Sunday
9/29/19	5:00 PM	207	Sunday
9/29/19	6:00 PM	186	Sunday
9/29/19	7:00 PM	204	Sunday
9/29/19	8:00 PM	156	Sunday
9/29/19	9:00 PM	38	Sunday
9/29/19	10:00 PM	0	Sunday
9/30/19	5:00 AM	7	Monday
9/30/19	6:00 AM	222	Monday
9/30/19	7:00 AM	177	Monday
9/30/19	8:00 AM	214	Monday
9/30/19	9:00 AM	364	Monday
9/30/19	10:00 AM	323	Monday
9/30/19	11:00 AM	275	Monday
9/30/19	12:00 PM	311	Monday

Figure 2: Organized data

If data is found in a **.pdf format**, you will need to manually convert the data into one of the ideal file types. Create an empty .csv or .xlsx file and import the numbers from the .pdf file one-by-one. Due to the high variation of .pdf files from varying sources, there is no perfect system that can automatically convert .pdf to a .csv or .xlsx file.

3.2 Web Scraping Data

If there is no option to download the data into a specific file format, you can use web scraping to collect the data directly off the site.

3.2.1 Scraping Data w/ BeautifulSoup

BeautifulSoup is a Python package that scrapes data off of websites based on the HTML website structure. Before diving into the package, it's important to understand the general structure of HTML.

To explore HTML, you can right click on the screen of any website and click "Inspect." Once you do so, it will pop up a panel that shows you the HTML structure of the site. Using your chosen website, you can then begin dissecting the HTML and using BeautifulSoup to locate specific tags to scrape. Since the data on websites may vary, feel free to look up tutorials like "How to scrape data off of Wikipedia with BeautifulSoup." For a general guide, we recommend watching this tutorial.

3.2.2 Scraping Social Media Data

There are a few packages in Python that can be used to scrape data from sites like Twitter and Reddit. For the purpose of this tutorial, we will be using SNScrape to collect tweets related to

Stoicism. SNScrape can also be used to scrape data from Reddit, Facebook, and Instagram. First, lets install the required packages. This line can be run in your computer's terminal, or in a Jupyter Notebook file.

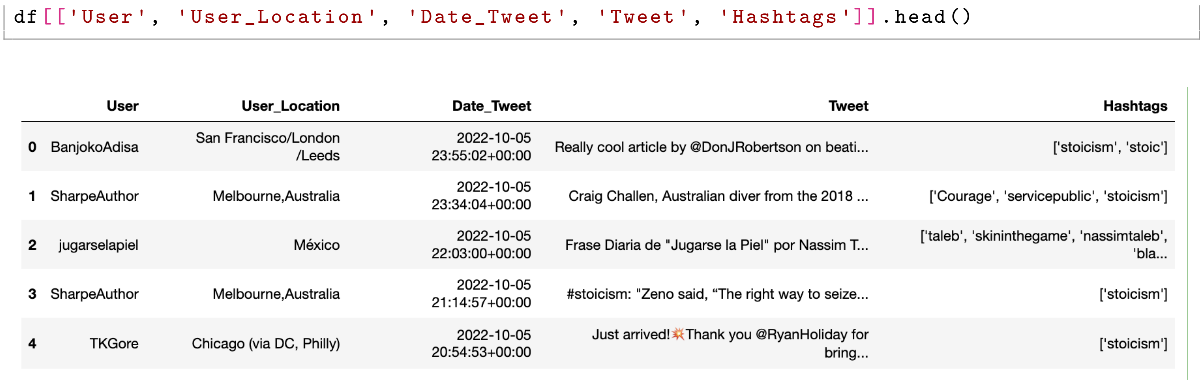
```
$ pip install pandas
$ pip install snscrape
```

Now lets import the packages into our file and get the scraper set up. The scraper will be defined in a function `tweets_scraper` that takes in a query and the number of desired tweets.

```python
import pandas as pd
import snscrape.modules.twitter as sntwitter

def tweet_scraper(query, n_tweet):
    attributes_container = []
    max_tweet = n_tweet

    for i,tweet in enumerate(sntwitter.TwitterSearchScraper(query).get_items()):

        if i>max_tweet:
            break

        attributes_container.append([tweet.user.username,
                                 tweet.user.verified,
                                 tweet.user.created,
                                 tweet.user.followersCount,
                                 tweet.user.friendsCount,
                                 tweet.user.location,
                                 tweet.retweetCount,
                                 tweet.lang,
                                 tweet.date,
                                 tweet.likeCount,
                                 tweet.sourceLabel,
                                 tweet.id,
                                 tweet.content,
                                 tweet.hashtags,
                                 tweet.conversationId,
                                 tweet.inReplyToUser,
                                 tweet.coordinates,
                                 tweet.place])

    return pd.DataFrame(attributes_container, columns=["User",
                                                   "verified",
                                                   "Date_Created",
                                                   "Follows_Count",
                                                   "Friends_Count",
                                                   "User_Location",
                                                   "Retweet_Count",
                                                   "Language",
                                                   "Date_Tweet",
                                                   "Number_of_Likes",
                                                   "Source_of_Tweet",
                                                   "Tweet_Id",
                                                   "Tweet",
                                                   "Hashtags",
                                                   "Conversation_Id",
                                                   "In_reply_To",
                                                   "Coordinates",
                                                   "Place"])
```

Depending on what information you want to get from the tweet, you can adjust the attributes in the `attributes_container` list within the function. If you do so, make sure that the `columns` parameter in the `return` statement reflects that change so it matches up. This is important as these will be the labels for the outputted Twitter data. To use the function, make sure all parameters (inputs) are in the proper format:

```python
df = tweet_scraper('#stoicism since:2018-07-05 until:2022-10-06', 10000)

# It's always important to view our data before diving into analysis
# Calling columns of interest in the brackets and .head() to see first 5 rows
```



```
df[['User', 'User_Location', 'Date_Tweet', 'Tweet', 'Hashtags']].head()
```

	User	User_Location	Date_Tweet	Tweet	Hashtags
0	BanjokoAdisa	San Francisco/London /Leeds	2022-10-05 23:55:02+00:00	Really cool article by @DonJRobertson on beati...	['stoicism', 'stoic']
1	SharpeAuthor	Melbourne,Australia	2022-10-05 23:34:04+00:00	Craig Challen, Australian diver from the 2018 ...	['Courage', 'servicepublic', 'stoicism']
2	jugarselapiel	México	2022-10-05 22:03:00+00:00	Frase Diaria de "Jugarse la Piel" por Nassim T...	['taleb', 'skininthegame', 'nassimtaleb', 'bla...
3	SharpeAuthor	Melbourne,Australia	2022-10-05 21:14:57+00:00	#stoicism: "Zeno said, “The right way to seize...	['stoicism']
4	TKGore	Chicago (via DC, Philly)	2022-10-05 20:54:53+00:00	Just arrived!💥Thank you @RyanHoliday for bring...	['stoicism']

Figure 3: Outputted dataframe of Stoicism tweets

3.3 Cleaning Data

Depending on the type of data you have, there will be a different process of cleaning the data. Cleaning data for social media text (like Twitter), article text (like Wikipedia pages), and numbers (like most popular Rec Cen times) will all result in slightly different methods. For example, if we're doing a text analysis on Twitter data, we might want to remove all emoji's and account handles to just look at the raw text tweet.

Cleaning your data before analyzing is extremely important. This ensure no biases or holes are in your data when you begin analyzing. Since most data we work with is numeric, a general rule of thumb when cleaning data is we want to ensure all NA or None values are removed. To make it easier for us to work with, it's also helpful to condense your data so that only the most relevant columns are there. For example, if we're analyzing the most popular Rec Cen times, it's not necessary to have column of data related to PERM numbers. Here's a general guide on cleaning data but feel free to search up more specific guides (ex. how to clean Twitter data in Python).

4 Analyzing and Visualizing Data

4.1 Important Principles of Statistics

Charts must follow statistical principles. It is unethical to modify data at any time to convey a particular point. If there is a bit of missing data or there are mistakes in the data (it happens), the appropriate way to note that would be to leave the graph unchanged and add a note or asterisk in the article or at the bottom of the chart explaining why the data is missing or incorrect. However, if there is a large amount of missing or incorrect data or any significant data piece missing or incorrect, it would be best to make a different graph that does not need the missing data.

Furthermore, note that complex graphs are not easily digestible for readers. Simplicity is important. Graphs should be easy to take in.

Basic Chart Types:

- For quantitative data (ex: change over time, correlation): line graphs (use dots to show points), bar graphs, area charts, scatter plots
- For qualitative data (ex: counts of items – i.e. number of each type of crimes): segmented bar charts, bar charts, pie charts

Sample Size:

- The required sample size does vary based on what the data is. If you are collecting a census (all possible data), any sample size is fine. For web scraping, 200+ texts are recommended. In general, based on what your population is, collect an appropriate amount of data that can reasonably represent the population.
- If you are missing a majority of your sample size, you cannot use the remaining data as representative of the whole. Find another sample.
- If you are comparing multiple groups with different sample sizes, you cannot compare the individual counts. You need to compare percentages instead. A segmented bar chart or pie chart is recommended.

Colors

- Colors on graphs must accurately represent the topic. Try to use colors that have some everyday relation to what you're graphing. A list of all traditional CSS Colors (you can use these color names in most coding software, including python) can be found here.
- Built-in color schemes for your software and color scheme generators are your best friend.
- If you don't like any traditional color or want to make your own color scheme, you can test hex codes and RGB values/build your own colors.

Association/Correlation

- Association and correlation are not causation. Do not imply them as such. Only replicated and randomized placebo-controlled experiments can prove causation.
- Make a scatter plot.
- Correlation describes the strength and direction of the linear relationship between an explanatory (x) and response (y) variable, indicated by the correlation coefficient, r — a number between -1 and 1.
- In order to accurately calculate the correlation coefficient, there needs to be a linear relationship between the variables, both variables should be normally distributed, and both variables should be quantitative. If any of those conditions are false, please do not calculate the correlation coefficient or R^2 (how much of the variation in y is explained by the differences in x). If you can calculate the correlation coefficient (all conditions are met) and there are outliers in the data, please remove the outliers and state in the article that you did and which ones you removed. Afterward, draw a line of best fit on the graph, but only if the plot of the residuals (google up how to graph a residual plot on your software) shows no pattern.

- If you cannot accurately calculate the correlation coefficient, describe the association between two variables instead. Association is, in general, the relationship between two variables. There is no numerical method nor conditions to describe association. Instead, describe the trends (form — straight, curved, etc.; direction — +/- slope; strength — how closely lines follow the form; unusual features — outliers/clusters) you see on the scatterplot.

4.2 Setting Up Coding Platforms

Google Colab:

This is a basic integrated development environment (IDE) to code in. It functions similarly to most google applications. The upside is that multiple people can collaborate on a single notebook (one code file), but note that they cannot work on the document at the same time. The downside is that it is incredibly slow and not updated with the latest tools. It is recommended to use other platforms, unless collaboration in a single notebook is essential to the project. You can access Google Colab here. If you want to access the latest version of the Python MatPlotLib package on Google Colab, please run the following code.

```
!pip install matplotlib --upgrade
!pip install -U matplotlib
```

Jupyter Notebook:

This is a powerful Python integrated development environment to code in. It is highly recommended and has a clean interface and runs code very quickly. To download it, first install Anaconda Navigator for your operating system. Open it and click Jupyter Notebook (not JupyterLab) and install it. To open Jupyter Notebook regularly after installation, just open Anaconda Navigator and open it from there. If you need additional help with installation, please read this guide for Windows and this guide for MacOs.

Note: Files that you use in your code (likely .csv or .txt files) should be uploaded to Jupyter Notebook in the same folder as your notebook.

R/RStudio:

RStudio is a powerful integrated development environment to code in, and it runs code very quickly. To install it, follow the instructions here. You will need to complete both steps. RStudio requires R to run.

Visual Studio Code:

Visual Studio Code is the most powerful integrated development environment to code in. It is one of the fastest IDEs out there, can support many coding platforms at once, and has many extensions and features. It can take a bit of time to become comfortable with the interface due to the plethora of tools available, but it is highly recommended to spend time learning how to use this software. Download here.

4.3 Exploring Data w/ Matplotlib

The Matplotlib Python package is a great way to explore data (see what patterns it reveals). In data science, this is called the Exploratory Data Analysis (EDA) process. Although effective for exploring and finding patterns, it's not the most visually pleasing. Hence, we have more tools below to create visually pleasing charts.

```
$ pip install matplotlib
```

```
# Every time we call the package we can simply call it by the shorthand
import matplotlib.pyplot as plt
import pandas as pd

# Create a sample data frame
data = {'x': [1, 2, 3, 4, 5], 'y': [2, 4, 6, 8, 10]}
df = pd.DataFrame(data)
```

```
# You'll probably have a .csv file to import instead of manually creating data here

# Create a line plot
df.plot(x='x', y='y', kind='line')

# Add a title and labels
plt.title('Line plot of x and y')
plt.xlabel('x')
plt.ylabel('y')

# Show the plot
plt.show()
```

There are a plethora of different charts you can create in Matplotlib, feel free to refer to the documentation or even better ask ChatGPT and it will more than likely output a feasible piece of code (ex. how to create a bar plot in Python matplotlib).

4.4 Creating Interactive Charts w/ Plotly

Graphing Charts:

Plotly is a chart creation tool, and it has both coding and non-coding options. You can code in Python, R, MATLAB, Javascript, and more. The documentation can be found here. Plotly is completely open-source, and you can find answers to questions in the documentation, on StackOverflow, and on the Plotly Forum.

The upside to plotly is that the charts are prettier than most tools, you can customize almost every aspect of the graph, and the charts are interactive. The downside is that the charts are not optimized for phone viewing, so if looking on a phone online at a Daily Nexus article with Plotly charts, it will look jumbled and messy on a phone. However, it will look beautiful in print and online on a computer. If you would like to explore creating Plotly charts that look good in print, online on a computer, and online on a phone, please look into Dash Plotly, which has the power to do that.

This repository contains a step-by-step code example on how to create line, bar, pie, and dropdown interactive charts with MatPlotLib, Plotly Express, and Plotly Graph Objects in Python. Plotly Express (known as "px" in the documentation) is mainly used to efficiently make simpler charts, while Plotly Graph Objects (known as "go" in the documentation) requires more manual code but allows for greater customization.

Please remove the blue background from the Plotly graphs (usually the data team uses white backgrounds, but for special-themed graphs feel free to use colorful backgrounds). Also, occasionally, you may want to remove the lines from a Plotly graph. The code to do that in Python is given below.

```
#hexcodes also work instead of color names
fig.update_layout(paper_bgcolor="white") #background color of everything but the graph
fig.update_layout(plot_bgcolor="white") #background color of graph
fig.update_xaxes(showgrid=False) #removes all vertical lines
fig.update_yaxes(showgrid=False) #removes all horizontal lines except the xaxis
```

Uploading Charts:

This code will upload your Plotly graph to Plotly chart studio and provide you with an iframe. Send that iframe to the data editors, who will paste it on the Daily Nexus's site. If you want to edit your charts, all you have to do is make the edit, and then re-run the py.plot code that corresponded to the graph. There is no need to resend the graph's iframe to the data editors for publishing. Thus, it is recommended to write each py.plot code in its own cell. For every graph you make, you will need a new filename, since otherwise Plotly may get confused.

You will need to make a Plotly account and make an API key by clicking your username in the top right-hand corner of your Plotly profile, then clicking settings, clicking API key, and then regenerate a new API key. Please save your API key for future use. You will need a new Plotly account after using up your 100 free charts.

```
!pip install chart_studio
import chart_studio
username = #make plotly account user
api_key = #make plotly account api key and regenerate
chart_studio.tools.set_credentials_file(username=username, api_key=api_key)
import chart_studio.plotly as py
import chart_studio.tools as tls
```

```
py.plot(insert_figure_name, filename = 'insert_file_name', auto_open=False) #This code
                                              will produce a link. Please copy that link
                                              .
```

The following code produces the iframe.

```
tls.get_embed('insert link produced from previous cell')
```

Please copy the iframe and remove the single quotes at the beginning and end. Please insert

```
 ?showlink=false
```

in the iframe immediately after the URL, without spaces. Please make sure the word “false” is lowercase. Don’t worry if the graph that is shown when you click the URL looks weird. The graph that will appear is the one in your code notebook.
This is how an iframe looks at first.

```
 '<iframe id="igraph" scrolling="no" style="border:none;" seamless="seamless" src="
                                              https://plotly.com/~yeahsidc/1.embed"
                                              height="525" width="100%"></iframe>'
```

This is how it should look after you edit it.

```
 <iframe id="igraph" scrolling="no" style="border:none;" seamless="seamless" src="
                                              https://plotly.com/~yeahsidc/1.embed?
                                              showlink=false" height="525" width="100%">
                                              </iframe>
```

Please now send your iframe to the data editors for publishing.

If your graph is not fitting on your computer when looking at the published article, you will need to change the height and width of the graph. Insert the following into your code and run the cell. Then rerun the corresponding py.plot box afterward. There is no need to resend the iframe.

```
insertfigurename.update_layout(height = insert_height_in_the_hundreds, width =
                                              insert_width_in_the_hundreds)
```

4.5 No-code Chart Tools

If you’re making a chart in a pinch, not too comfortable with code yet, or working with simple data, feel free to try visualizing data with some no-coding chart tools. Datawrapper allows you to input a .csv file and it will automatically turn your data into a chart. The benefit of this tool is how quickly you can generate a publishable chart, the downside is the limited flexibility.

On the other hand, if you have experience with graphic design or interested in creating more unique charts, feel free to check out Adobe or Figma. Adobe applications cost money but the Daily Nexus office has computers with paid subscriptions you can use. Figma is a free online tool that shares many similar features with some Adobe products. Using these tools, you can manually create each element of your data visualization.

4.6 Chart Grammar

- Multi-Year Format Example: Use 2020-21, not 2020-2021

  Example of how to change that in Plotly (since it auto formats years):

```
#Plotly Express
tick_vals = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] #start at 0 and
                                            continue until the end of how
                                            many years you have
tick_text = ['2011-12', '2012-13', '2013-14', '2014-15', '2015-16', '2016
                                            -17', '2017-18', '2018-19', '
                                            2019-20', '2020-21', '2021-22'
                                            ]
fig.update_layout(
xaxis={"tickmode":"array", "tickvals":tick_vals, "ticktext": tick_text}
)
```

```
#Plotly.graph_objects
fig.update_layout(
    xaxis = dict(
        tickmode = 'array',
        tickvals = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
        ticktext = ['2011-12', '2012-13', '2013-14', '2014-15', '2015-16'
                                                    , '2016-17', '2017-18'
                                                    , '2018-19', '2019-20'
                                                    , '2020-21', '2021-22'
                                                    ]
    )
)
```

- Graph titles should be bolded, centered, and in black font. It should be the same font size and font type as all other titles in the article. A common way to bold text is to use the below command (no spaces), which works in some, but not all software.

```
"<\b>2021-22 UCSB Men's Soccer Goals Scored By Player"
```

- All major words (i.e. not “the, ”that“, ”in“, etc.) should be capitalized in the title, axis titles, and elsewhere present in the graph. If the graph is interactive, please make sure that the hover text is formatted correctly and makes logical sense.

- Chart captions should be insightful and written as an action describing the message/impact of the chart, somewhat like a headline. It should not be a literal description of the chart — that is what the title is for.

5 Conducting Interviews

Interviews are an opportunity to find and/or solidify the story of an article. Interviewing at least 2 sources is, in most cases, mandatory. If you can't find at least 2 sources, please talk to the data editors, who can provide guidance. There are occasional exceptions such as some feature articles (1 interview) and some purely informational articles summarized from online sources (0 interviews), but even in those cases, if there is an opportunity to include additional interviews that add a meaningful piece to the article please do so. If there are more than 2 sides/perspectives to an article, please interview enough sources to cover all perspectives. All sides should have a fair chance to tell their story.

You must record all interviews. You also must receive permission from your source to record before you actually record (ask them once before the recording starts and then confirm permission live on the recording), since that is the law. Please try not to give anonymity unless it is incredibly (very high bar) sensitive material. Otter.ai is a free (as long as you keep resetting it/creating another count after using your free minutes) transcription app. However, you still have to manually listen and fix all transcription errors, because there will be a lot.

You cannot zone out during an interview. You must listen attentively and show respect. It is recommended to type out a list of questions beforehand and bring them on your computer to your interview, but please ask follow-up questions based on the source's responses. Do not stick purely to your script. Furthermore, while generally it is not a good idea to interrupt your sources, if you hear something that needs a follow-up, is incorrect information, or starts to go off-track, please respectfully ask about it. Don't be scared of your source — you are the interviewer. You're in power; If something needs calling out, do it. Sources are usually adults and don't need to be coddled, but should still be treated with respect.

Read this wonderful guide to learn how to prepare for and conduct an interview!

5.1 Who To Interview

- You CANNOT interview anyone you have some sort of professional relationship, romantic relationship, or friendship with. That is a conflict of interest and results in inaccurate interviews. Furthermore, to avoid bias, you should not be writing an article that involves a group or a person that you know because there is a high chance the reporting will have some sort of bias. Let someone else take the article!
- Think about who can be a representative of a perspective of the story. Usually, someone in a certain UCSB department is one of them. Google up the UCSB department you are interested in and see if you can find someone who matches your criteria or someone who deals with PR or Media. If you cannot find someone in that department, just send a member of the department the interview request template anyway and they will likely connect you to someone who can speak on the matter.
- Students can also be great sources. Students may work in departments, and you can use google to find someone to interview per the above bullet point. Feel free to ask your friends to give you potential sources to interview (you cannot interview your friends, however).
- Professionals can be informative sources as well. Google can help you find their contact information, and you can set up a zoom or phone interview with them.
- Feel free to cold email and follow up with potential sources. Many of them may not respond to you, but some eventually will!
- For general UCSB matters (use to get an official UCSB perspective) and as a last resort when you can't find anyone to interview, e-mail the UCSB Media Relations Manager, Kiki Reyes. She can be reached at kikireyes@ucsb.edu. Note: In your interview request, send her your questions via email and state your deadline for when you want the response, and she will respond back in an email statement. (See the Kiki Reyes Template below).

5.2 Interview Request Template

- **General:**

  Subject: Daily Nexus Interview Request About [**insert topic**]

  Hi [**insert name**],

  My name is [**insert name**], and I'm a reporter with the Daily Nexus — the independent, student-run newspaper at UC Santa Barbara.

  I'm currently working on an article regarding [**insert topic**]. I was hoping to interview you about your [**insert why you're interviewing them**] for this article.

  I'm free on [**insert days/times**]. Would any of these times work for you, and do you have a preferred location?

  Thank you so much for your time, and I look forward to speaking with you.

  Best,

  [**insert full name**]

  Daily Nexus reporter

- **Kiki Reyes:**

  Subject: Daily Nexus Interview Request About [**insert topic**]

  Hi [**insert name**],

  My name is [**insert name**], and I'm a reporter with the Daily Nexus — the independent, student-run newspaper at UC Santa Barbara.

  I'm currently working on an article regarding [**insert topic**], and I would like to interview you about [**insert why you're interviewing them**]. I would appreciate a response by [**insert date**] to meet my deadline for the article. Could you please answer the following questions?

  [**insert questions**]

  Thank you so much for your time.

  Best,

  [**insert full name**]

  Daily Nexus reporter

- **Professional Not at UCSB:**

  Subject: Daily Nexus Interview Request About [**insert topic**]

  Hi [**insert name**],

  My name is [**insert name**], and I'm a reporter with the Daily Nexus — the independent, student-run newspaper at UC Santa Barbara.

  I'm currently working on an article regarding [**insert topic**]. I was hoping to interview you about your [**insert why you're interviewing them**] for this article.

  I'm free on [**insert days/times**]. Are you available during any of those times for a zoom interview?

  Thank you so much for your time, and I look forward to speaking with you.

  Best,

  [**insert full name**]

  Daily Nexus reporter

5.3 Interview Structure

- Before you start the interview, you must ask, “Do I have permission to record this interview?”. Once the source agrees, start the recording and ask, “To confirm on record, do I have permission to record this interview?”. Note: If the source requests to go off the record at any point, you may not use any information from the off-record portion in your article directly.

- If the source is a student, ask them for their full name, major, year, and pronouns. If the source is not a student, ask them for their full name, job title, and pronouns. This information will need to be included in the article.

- Do not jump directly to the main reason why you’re interviewing them. You first need contextual information and to gain your source’s trust. Ask them about their background (i.e. where did they grow up, what inspired them to do whatever it is that you’re interviewing about, etc.)

- Now, ask them about the main topic. Make sure to listen attentively and ask follow-up questions. Spent a large portion of the interview here.

- After gathering enough information, ask them about the impact, of whatever it is that you are interviewing about, on your source. Ask them what they believe the impact will be for the greater community as well. The impact is the most important part and where stories are built.

- Once you have finished all your questions, ask them, “Do you have anything else you would like to share?”. Afterward, thank them for your time. When the article is published, please send your source your finished article. Feel free to ask clarifying questions to them afterward via email. Do not send them any draft of the article before publishing, and do not allow them to influence any part of your article. If they try to control what you publish, except for off-record requests, please decline and state that you are not allowed to do that.

5.4 Incorporating Interviews in Articles

- After transcribing your interview, search the transcript for a couple of quotes that **ADD** information or perspective to the article. Do not use the entire interview. An article should not just be quotes from one person.

- Decide if a summary is appropriate, or a quote. Usually, a combination where you summarize something that the source said and then include a quote afterward is common.

- All summaries and quotes must be correctly attributed to the information you got at the start of the interview (full name, job title/major, year if student). If you quote something, you must type out the quote exactly as it is said. Do not use the entire quote. Small edits for grammar and punctuation are allowed. If you absolutely need to clarify something in the quote, use brackets. Try wherever possible not to use brackets.

  Example Quote: “They are my favorite candies.”

  Correct Usage: “[Skittles and Almond Joy] are my favorite candies.”

  Wrong Usage: “They [Skittles and Almond Joy] are my favorite candies.”

  Example Quote: “Last week, I went to Albertson’s, and there I saw Skittles and Almond Joy for the first time. I was immediately struck by their beauty, and I piled up as many as I could into my shopping cart.” - *Bob Smith, student*

  Correct Usage: Bob Smith, a third-year history major at UCSB, saw Skittles and Almond Joy for the first time during a trip to Albertson’s last week and was galvanized by their beauty. “I piled up as many as I could into my shopping cart,” he said.

Wrong Usage: “Last week, I went to Albertson’s, and there I saw Skittles and Almond Joy for the first time. I was immediately struck by their beauty, and I piled up as many as I could into my shopping cart.” said Bob Smith.

Example Quote: “I believe that music has the potential to change the world. Each note represents an element of a story, which comes together at the score’s conclusion. Since concertgoers are surrounded by other people dancing to the music, and thus enjoying the storyline, they can directly see that the music has a large impact on others. As such, when the audience leaves a concert, they are left with a sense of purpose and motivated to improve their communities.” - *Greg Michael, concertgoer. Assume the reporter has already cited Michael somewhere else in the article and wants to close the article with a quote.*

Correct Usage: “I believe that music has the potential to change the world,” Michael said. “When the audience leaves a concert, they are left with a sense of purpose and motivated to improve their communities.”

Wrong Usage: “I believe that music has the potential to change the world. Each [music note] represents an element of a story, which comes together at the score’s conclusion.” Michael said, “when the audience leaves a concert, they are left with a sense of purpose and motivated to improve their communities.”

- This is journalism, not literature class. Quotes do not need to be integrated, but they can be as long as the article remains clear.
- You do not need to mention in the article that the so-and-so said this “in an interview”. The exception is if you gathered the information via an email statement. Then, you say this: “According to John Prowler in an email statement to the *Nexus*” (“to the *Nexus* is optional”)
- Punctuation, including periods, almost always goes inside the quote.
- Try not to use this format if possible, but tf using the format–Turner said, “quote quote quote.”– then the first word of the quote is capitalized only if the quote is a complete sentence. There is always a comma after the speaking verb (i.e. “said”).
- For data, a common way to end the article is with a quote from an interview or a summary of findings. If using a quote, make sure it is a quote that touches on impact. The structure of the last example of the three examples above is a great method to close.

6 Writing the Article

6.1 What Story Does The Data Tell?

The focus of a data article should be on the story. Data are values along with context, and numbers should be used to convey the topic of the article. Many readers prefer easily digestible stories rather than staring at a bunch of numbers or endless analysis. Human attention spans are short. With this in mind, keep articles concise and pick and choose just the main statistics from a graph to analyze. Trust your graph to do its work.

6.2 Structure

Headline

The headline is a couple of words that inform the reader what the news is. They should be objective and somewhat concise. Only the first word and proper nouns in headlines should be capitalized. Headlines should not merely hint at the story — the Daily Nexus does not partake in clickbait. The Daily Nexus News Team has a great collection of headlines on the website which serve as examples on how to make a headline. An example is also listed below.

> **Article Topic:** The UCSB baseball team' just won its conference. You are analyzing the team's record, stats, and progress so far.
>
> **Correct Headline Formation:** Powered by bunts and disciplined defense, UCSB men's baseball team wins conference
>
> **Incorrect Headline Formation:** A look into the UCSB men's baseball team
>
> **Incorrect Headline Formation:** UCSB's great, amazing, and hard-working baseball team caps off season with an invigorating conference title game victory

Constructing the Article

Read this document and Ctrl+F or Command+F "Inverted Pyramid News Style". Please note the data-specific things below. The rest of that document is also an incredibly helpful tool to learn how to write articles, so please read it in its entirety. Construct your own article in google docs, unless it is a pure graphic design visualization, so that it is shareable.

Data-Specific Things

- You could briefly summarize findings in the lede/nutgraph, in a results section, or at the end in a conclusion. It is alright to add a conclusion section for data articles, if you would like.
- Occasionally, depending on what you're writing about, you may want to break up your article into sections like this article does.
- The guide above states that articles are written in a transition/lead-in and then quote format. Data articles are written in a transition/lead-in, quote/chart/data, and then analysis format.

Hyperlinks

Anytime you reference a source online, refer to an entity that has an online presence, or believe that your article could be improved with additional information from an outside source, please hyperlink that source in your article. You should select either a verb or a noun that is no longer than 1-2, potentially 3 words (except for titles of organizations, which can have as many words as are in the title) and press Ctrl+K or Command+K in Google Docs. Then, insert the link to the webpage that you want to be hyperlinked. Refer to all the blue, hyperlinked words in this guide as an example of hyperlink format.

6.3 Grammar and AP Style 101

- Proofread. Proofread. Proofread. Please! Looking back at your work and double-checking with Google and Grammarly is free and worth your time.
- Only capitalize proper nouns (just as you would do in an essay).
- Punctuation almost always goes inside the quotes.
- A complete sentence has a subject, verb, and a complete thought.
- When writing a compound sentence (Ex: I went to the store, and I bought flowers). There should always be a comma before the conjunction. When using a compound verb (Ex: I went to the store and bought flowers), there should almost never be a comma before the conjunction.
- Do not use the oxford comma (the last comma before "and" in a series of items).
- In most cases, spell out numbers 1-9 and use numerals for the rest.
- Quote formatting: "insert quote", John said. Do not do this: "insert quote", said John.

6.4 AP Style Guide

The Daily Nexus follows the Master AP Style Guide.

6.5 Common Errors to Look Out For

- Don't write paragraphs longer than 1-3 sentences long. Change paragraphs with each following story element so readers aren't bombarded with too much info to digest at once.
- Don't throw a bunch of statistics at your reader. This is journalism and not a data report. Choose one or two main points and explain their context and impact, otherwise, a lay (not well-versed) reader may just skip over your work. Why do those numbers matter?
- Don't ask questions. Your job as the reporter is to answer the questions.
- Don't be informal. Use complete sentences and maintain professionalism.
- Don't use personal pronouns or possessives like "I", "my", "we", "our", "us", etc. Data journalism should be objective and focus on telling the story, not advocating for or against it.
- Don't use acronyms or refer to a person's last name without stating the full name at least once prior. Afterwards, it is completely fine.

7 Resources

Google is your number one resource. Almost everything can be found on google. Sometimes, you have to search multiple times (and take a couple of breaks) before you find the answer to a specific problem, but there is a very high chance that you will find your answer. **For coding:**

- YouTube: searching up tutorials that match your task
- Stack Overflow: searching any coding question on Google will often surface a StackExchange forum. Most of the time, your question will likely have been answered by someone. If not, feel free to ask the question in the forum.
- Documentation: most coding packages will have its own documentation written by the creators of the package. Leverage this resource to understand the basics of the various functions a package offers.
- Data Team Github: most of our past projects have code that is public on our Github. Feel free to explore and see how we've used various tools (Python, R, JavaScript, etc) for data projects.
- A guide to coding graphs in Javascript's D3.js, one of the most advanced and commonly used platforms in data journalism.
- An app to scrape Discord texts and see all channel names, even hidden ones (it can only scrape channels you are a part of however). You do not need admin permissions, and no one will be notified.

For writing:

- Daily Nexus Print: past print articles can be found there. This is particularly helpful if you're interested in seeing the layout of the newspaper and planning how you want your charts to be placed.
- Other Data Publications: some notable data publications include UCLA's The Stack and UCLA's Bruin Sports Analytics. These are great to reference for new ideas, structuring articles, etc.