[Bio 1B] Bioindicators of Strawberry Creek

Professors George Roderick, John Huelsenbeck & Alan Shabel

Estimated Time: *50 minutes*
Notebook Created by: Andy Sheu, Joshua Asuncion and Karalyn Chong
Code Maintenance: Elias Saravia

Welcome! Throughout this lab you will use Python to analyze the data collected from the North and South Forks of Strawberry Creek. Python is a general-purpose programming language that allows one to use data analysis methods to answer questions about data. In this part of the lab we will apply statistical methods to the biological metrics data to determine whether or not the water quality (or ecological health) of the two forks of Strawberry Creek is significantly different.

Learning Outcomes

By the end of this notebook and accompanying lab, students should be able to:

1. Explain the use of biological organisms as indicators of ecosystem health.
2. Interpret biological metrics of diversity: taxon richness, %EPT, biotic index (FBI), % filterers, % predators, Shannon index.
3. Understand how to construct a Null and Alternative Hypothesis.
4. Use randomization to determine if two distributions are different.
5. Interpret a p-value to describe statistical significance.

Table of Contents

1. Jupyter Notebooks

This portion of the lab is set up in a Jupyter Notebook. A Jupyter Notebook is an online, interactive computing environment, composed of different types of **cells**. Cells are chunks of code or text that are used to break up a larger notebook into smaller, more manageable parts and to let the viewer modify and interact with the elements of the notebook.

Types of cells

There are two types of cells in Jupyter, **code** cells and **markdown** cells. Code cells are cells indicated with “In [ ]:” to the left of the cell. In these cells you can write you own code and run the code in the individual cell. Markdown cells hold text a majority of the time and do not have the “In [ ]” to the left of the cell.

Running cells

'Running' a cell is similar to pressing 'Enter' on a calculator once you've typed in an expression; it computes all of the expressions contained within the cell.

To run a code cell, you can do one of the following:

- press **Shift + Enter**
- click **Cell -> Run Cells** in the toolbar at the top of the screen.

You can navigate the cells by either clicking on them or by using your up and down arrow keys. Try running the cell below to see what happens.

In [ ]:
```
print("Hello, World")
```

The input of the cell consists of the text/code that is contained within the cell's enclosing box. Here, the input is an expression in Python that "prints" or repeats whatever text or number is passed in.

The output of running a cell is shown in the line immediately after it. Notice that markdown cells have no output.

Editing, Saving and Submitting

- To **edit** a cell simply click on the desired cell and begin typing
- To **save** your notebook press *command + s* on the keyboard
- We will go into the specifics of how to **submit** your work at the end of the lab, but you will essentially be converting your work into a PDF file and then including it in your Lab Report

Run this cell before proceeding with the rest of the lab!

In [1]:
```
from otter import Notebook

import numpy as np
import pandas as pd
from strawberry_creek_widget import *
import ipywidgets as widgets
from ipywidgets import interact, interact_manual, fixed
from IPython.display import display
from IPython.display import clear_output
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline
```

2. Importing the Data

In this section you will import the data that you collected in the lab!

To import your data you must:

1. Download the combined class data from Google Drive and save to you computer's desktop as a .csv file with the name : strawberry_creek.csv
2. Click on the `Jupyterhub` icon on the top left hand corner. You will be redirected to another page.
3. Next, open the `Bio-1B >>> Strawberry Creek` folder. This is where you will be uploading your data.
4. In the top right hand corner, click `Upload` and select the file name `strawberry_creek.csv` . Once you upload the data, return to this notebook titled `Bioindicators Notebook.ipynb` .
5. In the code cell below, replace `"SC_data.csv"` with `"strawberry_creek.csv"` and your data should load.

> **Note:** If you are unable to upload the class combined data, please contact your lab instructor for the data file. As a last resort, you can let your instructor know that you will use the default data file: `"SC_data.csv"` for this notebook.

To import the data set just run the following cell! If all goes smoothly, you will see the first few rows of your data file.

In [2]:

Out[2]:

	Group	Fork	Richness	EPT	FBI	Filters	Predators	Shannon
0	3	North	8	35	5.48	52	12	1.942
1	4	North	9	34	5.72	58	12	1.680
2	5	North	10	32	5.42	50	14	1.950
3	6	North	10	32	5.40	48	4	1.930
4	9	North	8	34	5.60	56	6	1.780

3. Introduction to Data Analytics

Null Hypothesis vs. Alternative Hypothesis

One of the first problems to work through when looking at a data set is to determine whether or not the trends in the data are significant or purely due to random chance. In this lab we are trying to determine whether or not the difference between communities of organisms in the two forks of the creek are different from one another or not. If we determine that our samples are more different than is expected by chance, then we can say that the difference is significant and conclude that our samples represent real differences in the communities of macroinvertebrates.

To do this we begin by forming a null hypothesis and an alternative hypothesis to test.

> **Null Hypothesis**: A null hypothesis claims that there is no statistical difference between two samples and that any difference is due to experimental error, measurement error, or chance.
>
> **Alternative Hypothesis**: An alternative hypothesis states that the difference in samples is meaningful, or significant.

Example Null and Alternative Hypothesis

Say we have a data set with samples of the number of boba shops on Southside and Northside. The data set shows that Southside has a higher average of boba shops than Northside, but it is unclear whether the difference in the average is due to chance or some other unknown reason. For this data set, potential hypotheses would be:

> **Example Null Hypothesis**
>
> - The distribution of the average number of boba shops is the same for the samples taken from Southside as the samples taken from Northside. The difference in sample distribution is due to chance.
>
> **Example Alternative Hypothesis**
>
> - The average number of boba shops in samples from Southside is lower than the average number of boba shops in samples from Northside.

Discussion Questions

Question 1a

What is your null hypothesis with regard to the relative water quality of the north and south fork of Strawberry Creek?

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

Question 1b

What is your alternative hypothesis?

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

After you have your null and alternative hypothesis, the next step is to simulate the distribution under the null hypothesis! Theoretically, if the differences in distributions were solely due to random chance, then the data that the distribution originally comes from would be the same. This is where randomization tests come in to play.

Randomization Test

A randomization test **shuffles a data set among categories and creates new distributions**. In this case, we are using a randomization test to shuffle the difference in ecological health of the two creeks. As was previously mentioned, a randomization test simulates the null hypothesis because it assumes that there is no significant difference between the distributions.

To demonstrate, we will run randomization testing on example data of a biotic index (FBI scores) collected from the North and South Fork in order to understand the process. You will analyze your own data after this.

Run the following code below to enter the example data and see it displayed in a data frame.

```
In [3]:
```

Out[3]:

	FBI Score	Fork
0	3.5	North
1	4.0	North
2	3.0	North
3	3.5	North
4	4.2	North
5	4.5	South
6	5.0	South
7	3.6	South
8	4.8	South
9	5.1	South
10	3.4	South
11	2.9	South

Here, we see each row of the dataset represents an `FBI Score` for a specific `Fork`, either North or South.

With the data, we can compare the North and South Fork by calculating the difference between the means of each Fork. Run the cell below to see the observed difference in FBI means between the two samples.

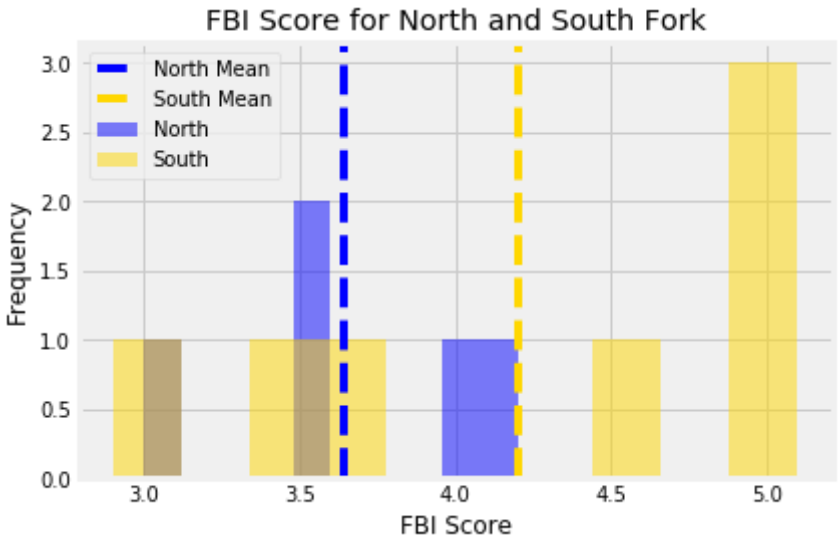
In [4]:

Out[4]:
```
FBI Score    0.56
dtype: float64
```

We call this our observed difference because this statistic is observed from data that was actually collected.

To illustrate how we find the observed difference, we plot the distribution of FBI scores for each Fork. For each distribution, we plot its mean. Notice that subtracting the two means results in our observed difference from above.

In [5]:



In randomization testing, the data points are 'shuffled' between the two forks. That is, the analysis takes data from the North Fork and the South Fork and creates a new data set by placing the data into new North and South Fork data sets **randomly**. In this way we can test how likely it is to obtain the observed differences between the North and South fork by chance alone.

For one randomization, we will calculate the FBI Score means for each fork. In this case, the mean difference is no longer an observed difference but a simulated difference. Run the cells below to generate a randomization of the data and to calculate the new difference.

In [6]:

Out[6]:

	FBI Score	Fork
2	3.0	North
9	5.1	North
7	3.6	North
6	5.0	North
10	3.4	North
8	4.9	South
4	4.2	South
1	4.0	South
11	2.9	South
0	3.5	South
3	3.5	South
5	4.5	South

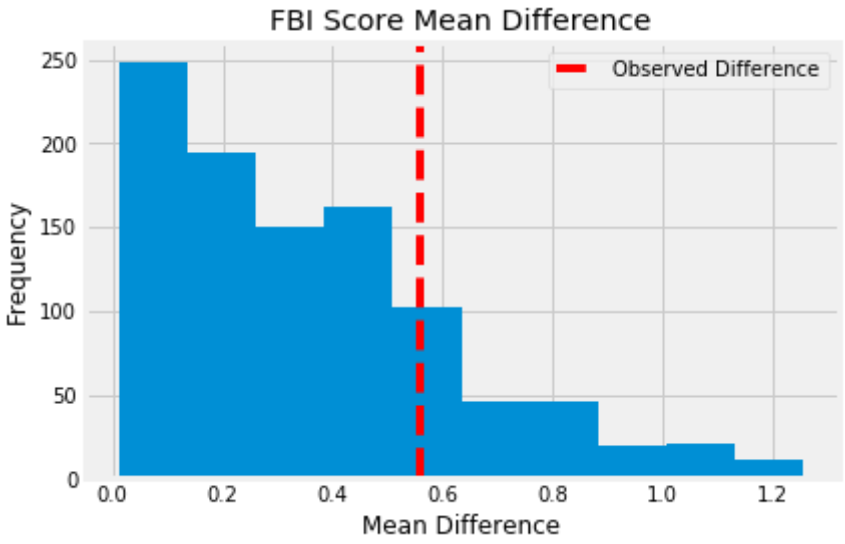
In [7]:

Out[7]:
```
FBI Score    0.091429
dtype: float64
```

This is just for one randomization of the data. We perform many randomization tests (about 1000 of them) and with these values we can plot the distribution of differences of means. Using this distribution of simulated differences, we can compare it with our actual observed difference to see how likely it is to observe this difference and if our null hypothesis is true.

In [8]:

In [9]:



This chart shows a distribution of differences (e.g. mean of north fork samples - mean of south fork samples) for 1000 randomized simulations. We can see that most often the difference between the means are below 0.6. The mean from the observed data is indicated by the dashed red line.

Using this plot, we can guess if the null hypothesis is true (the observed difference between the two forks is due to random chance) or if the alternative hypothesis is true (that it is not due to chance alone).

Discussion Question

Question 2

For the FBI metric, how likely is it for the observed difference to occur, and can we reject the null hypothesis?

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

P-Values & Statistical Significance

Now that we have a distribution of what the differences in FBI Scores generally look like, we can calculate the p-value to determine how probable it is for the observed_difference to occur. To calculate the p-value we count the number of times the difference is more extreme than the observed difference in the distribution and divide it by the total number of randomizations.

```
In [10]:
```

```
Out[10]: 0.201
```

If the p-value is small, it indicates that it is very unlikely for this result to occur and we say we "reject the null hypothesis", meaning that the observed data likely represent an actual difference between the North and South Fork samples. Otherwise, if the p-value is large, it implies that the observed test statistic has a high likelihood of occurring and we say we "fail to reject the null hypothesis".

A conventional cut-off for p-values is 0.05 or 5%. If the p-value is **less than or equal to 5%**, then the p-value is deemed **"statistically significant"**. Here, the p-value is larger than that. We will discuss p-values more in lab.

Discussion Question

Question 3

Using the calculated p-value above, do we reject the null hypothesis or fail to reject the null hypothesis? Why?

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

4. Your Data

Now, instead of using example data, you will use the data you imported and calculate the mean differences for each of the metrics you measured. Run the next cell for the observed differences between forks for each of the biological measures.

In [11]:

Out[11]:
```
Group         0.00000
Richness      0.20000
EPT          18.10000
FBI           2.22200
Filters      26.60000
Predators     1.40000
Shannon       0.05401
dtype: float64
```

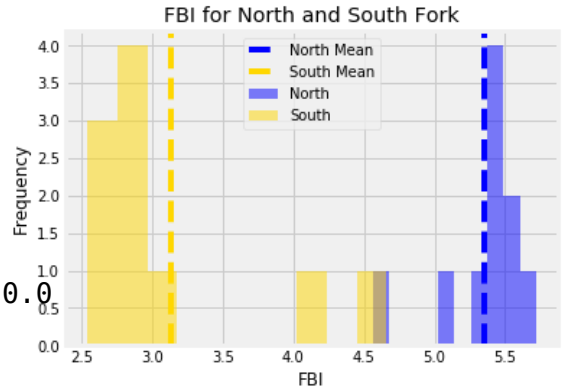
Let's start with the FBI index. The next few cells run the randomizations, graph the data, and calculate a p-value, just like for the sample data above.

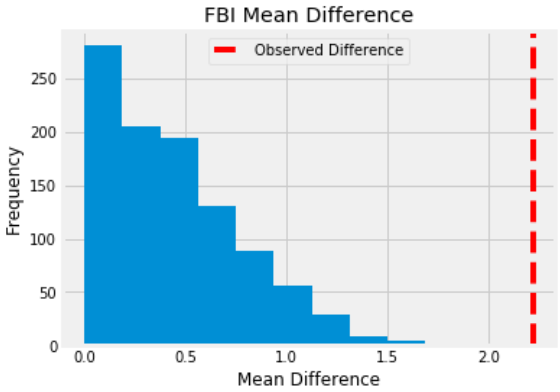
In [12]:

In [13]:

In [14]:

Out[14]: 0.0





Now, it is easy to repeat the same analysis for each of the other biological measures you collected in the lab. One of the values of using a Notebook is that it makes repeating the same process easy. For instance, we can use a widget that takes the code and makes it interactive, so that you can select in a dropdown menu which metric you want to use.

Run the following code cell to instantiate the interactive widget. Try selecting different metrics to see how the plot changes. You will also see that on the right side, we have plotted a histogram of the metric split by the two Forks.

In [15]:

Discussion Questions

Question 4a

For each of the options in the dropdown menu (FBI Index, Richness, %ETP, %Filters, %Predators and Shannon Index), explain what the histograms mean and why the histograms look the way they do.

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

Question 4b

Do you see anything unusual about any of the histograms or are they consistent with your expectations and hypotheses?

WRITE YOUR ANSWER HERE. REPLACE THIS LINE WITH YOUR ANSWER BY DOUBLE-CLICKING THE CELL.

5. Submitting the Lab

Submitting your work

Run the code cell below convert your answers to the discussion question into a pdf. Be sure to follow instructions for submitting this assignment.

- After running the cell, right-click on `Download it here!` then click `Save Link As...` to save it as a PDF.

In [ ]:

In [2]:

```
absl-py==0.9.0
aiohttp==3.6.2
alabaster==0.7.12
alembic==1.4.2
allensdk==0.16.0
appdirs==1.4.4
appmode==0.4.0
arviz==0.8.2
astor==0.8.1
astropy==4.0
astroquery==0.3.10
async-generator==1.10
async-timeout==3.0.1
attrs==19.3.0
Babel==2.8.0
backcall==0.1.0
basemap==1.1.0
beautifulsoup4==4.6.0
bitarray==0.8.3
bkcharts==0.2
bleach==3.1.5
bokeh==0.12.6
branca==0.4.1
brewer2mpl==1.4.1
cachetools==4.1.0
Cartopy==0.17.0
certifi==2020.4.5.1
certipy==0.1.3
cffi==1.14.0
chardet==3.0.4
chart-studio==1.0.0
click==7.1.2
click-plugins==1.1.1
cligj==0.5.0
cloudpickle==1.1.1
corner==2.0.1
coverage==4.5.3
coveralls==2.0.0
cryptography==2.9.2
csaps==0.5.0
cycler==0.10.0
cymem==1.31.2
Cython==0.29.19
cytoolz==0.8.2
daft==0.0.4
dask==2.3.0
datascience==0.15.1
decorator==4.4.2
defusedxml==0.6.0
descartes==1.1.0
dill==0.2.9
docker==4.2.0
docopt==0.6.2
docutils==0.16
dustmaps==1.0.4
emcee==2.2.1
entrypoints==0.3
```

```
et-xmlfile==1.0.1
exoplanet==0.2.4
ffmpeg-python==0.1.17
Fiona==1.8.13.post1
folium==0.9.1
ftfy==4.4.3
future==0.18.2
gast==0.2.2
geojson==2.5.0
geopandas==0.3.0
geopy==1.11.0
george==0.3.1
GetDist==0.3.3
ggplot==0.11.5
gmaps==0.8.0
google-auth==1.15.0
google-auth-oauthlib==0.4.1
google-pasta==0.2.0
gradable-nbexport==0.0.2
grpcio==1.29.0
gspread==3.1.0
gspread-pandas==1.2.1
h5py==2.7.0
hdbscan==0.8.22
healpy==1.13.0
hmms==0.1
html5lib==1.0.1
httplib2==0.18.1
idna==2.8
idna-ssl==1.1.0
imagesize==1.2.0
importlib-metadata==1.6.0
ipykernel==5.3.0
ipympl==0.2.1
ipython==7.14.0
ipython-genutils==0.2.0
ipywidgets==7.2.1
jdcal==1.4.1
jedi==0.17.0
jeepney==0.4.3
Jinja2==2.11.2
joblib==0.15.1
json5==0.9.5
jsonschema==3.2.0
jupyter-client==6.1.3
jupyter-core==4.6.3
jupyter-rsession-proxy==1.0b6
jupyter-server-proxy==1.0.1
jupyterhub==1.0.0
jupyterlab==2.1.3
jupyterlab-server==1.1.5
Keras==2.0.8
Keras-Applications==1.0.8
Keras-Preprocessing==1.1.2
keras-vis==0.4.1
keyring==21.2.1
kiwisolver==1.2.0
```

```
lcapy==0.44.1
llvmlite==0.32.1
lxml==3.8.0
Mako==1.1.2
Markdown==3.2.2
MarkupSafe==1.1.1
matplotlib==3.1.0
mcautograder==0.0.5
mistune==0.8.4
mne==0.14.1
more-itertools==8.3.0
mplleaflet==0.0.5
mpmath==1.1.0
multidict==4.7.6
munch==2.5.0
murmurhash==0.26.4
nb2pdf==0.5.0
nbconvert==5.6.1
nbformat==5.0.6
nbforms==0.5.1
nbgitpuller==0.7.2
nbpdfexport==0.2.1
nbresuse==0.3.2
nbzip==0.0.4
netCDF4==1.3.1
networkx==2.4
nibabel==2.1.0
nilearn==0.3.1
nlmpy==0.1.5
nltk==3.4.5
nose==1.3.7
notebook==5.7.8
numba==0.49.1
numexpr==2.6.4
numpy==1.16.0
oauth2client==4.1.3
oauthlib==3.1.0
okpy==1.12.5
olefile==0.46
opencv-python==3.3.0.10
openpyxl==2.4.8
opt-einsum==3.2.1
otter-grader==1.0.0
packaging==20.4
pamela==1.0.0
pandas==0.23.4
pandocfilters==1.4.2
parso==0.7.0
pathlib==1.0.1
patsy==0.5.1
pexpect==4.8.0
pickleshare==0.7.5
Pillow==4.2.1
Pint==0.9
pkg-resources==0.0.0
plac==0.9.6
plotly==4.0.0
```

```
pluggy==0.13.1
preshed==1.0.1
progressbar2==3.51.3
prometheus-client==0.8.0
prompt-toolkit==3.0.5
protobuf==3.12.1
psutil==5.7.0
ptyprocess==0.6.0
py==1.8.1
pyasn1==0.4.8
pyasn1-modules==0.2.8
pybind11==2.5.0
pycparser==2.20
pydot==1.4.1
pyee==7.0.2
pygame==1.9.3
Pygments==2.6.1
pymc3==3.8
pymdptoolbox==4.0b3
pynrrd==0.4.2
pyOpenSSL==19.1.0
pyparsing==2.4.7
pyppeteer==0.2.2
pyproj==2.6.1.post1
pyreadstat==0.2.8
pyrsistent==0.16.0
PySAL==1.14.3
pyshp==2.1.0
pytest==5.4.2
python-dateutil==2.8.1
python-editor==1.0.4
python-utils==2.4.0
pytz==2020.1
PyVCF==0.6.8
PyWavelets==1.1.1
PyYAML==5.3.1
pyzmq==19.0.1
qgrid==1.1.1
regex==2017.11.9
requests==2.21.0
requests-oauthlib==1.3.0
requests-toolbelt==0.9.1
retrying==1.3.3
rsa==4.0
Rtree==0.8.3
scikit-image==0.13.1
scikit-learn==0.21.3
scipy==1.2.0
seaborn==0.9.0
SecretStorage==3.1.2
Send2Trash==1.5.0
Shapely==1.6.4.post2
simpervisor==0.3
SimpleITK==1.2.4
simplejson==3.17.0
six==1.15.0
snowballstemmer==2.0.0
```

```
spacy==1.9.0
Sphinx==3.0.4
sphinxcontrib-applehelp==1.0.2
sphinxcontrib-devhelp==1.0.2
sphinxcontrib-htmlhelp==1.0.3
sphinxcontrib-jsmath==1.0.1
sphinxcontrib-qthelp==1.0.3
sphinxcontrib-serializinghtml==1.1.4
SQLAlchemy==1.3.17
statsmodels==0.8.0
sympy==1.4
tables==3.4.2
tensorboard==2.1.1
tensorflow==2.1.0
tensorflow-estimator==2.1.0
tensorflow-hub==0.7.0
tensorflow-probability==0.8.0
termcolor==1.1.0
terminado==0.8.3
testpath==0.4.4
Theano==1.0.4
thinc==6.5.2
toolz==0.10.0
torch==1.3.1
torchvision==0.2.2
tornado==5.1.1
tqdm==4.15.0
traitlets==4.3.3
typing-extensions==3.7.4.2
ujson==2.0.3
umap-learn==0.3.10
urllib3==1.24.3
wcwidth==0.1.9
webencodings==0.5.1
websocket-client==0.57.0
websockets==8.1
Werkzeug==1.0.1
widgetsnbextension==3.2.1
wordcloud==1.5.0
wrapt==1.12.1
xarray==0.12.3
yarl==1.4.2
zipp==3.1.0
You are using pip version 9.0.3, however version 20.1.1 is availabl
e.
You should consider upgrading via the 'pip install --upgrade pip' co
mmand.
```

Feedback Form

Please fill out this form (https://forms.gle/AuDrBiJDDyP8ktzUA) to give us valuable feedback for later notebooks!

Data Science Opportunities

Data Science Modules: http://data.berkeley.edu/education/modules (http://data.berkeley.edu/education/modules)

Data Science Offerings at Berkeley: https://data.berkeley.edu/academics/undergraduate-programs/data-science-offerings (https://data.berkeley.edu/academics/undergraduate-programs/data-science-offerings)

In [ ]: