DS002.3.2 Python Review 2

Start with another poll, https://forms.gle/Mn9r8ZZA6K7WeLY19

Admin

1. Move from Deepnote to Colab
2. Cloning code from GitHub
3. You can't easily push changes to GH from Google Colab
4. Deepnote, Colab, and Jupyter notebooks have all the same stuff
5. Today's homework

Review

Colab notebook

Pyplot

Importing code from GitHub into Colab

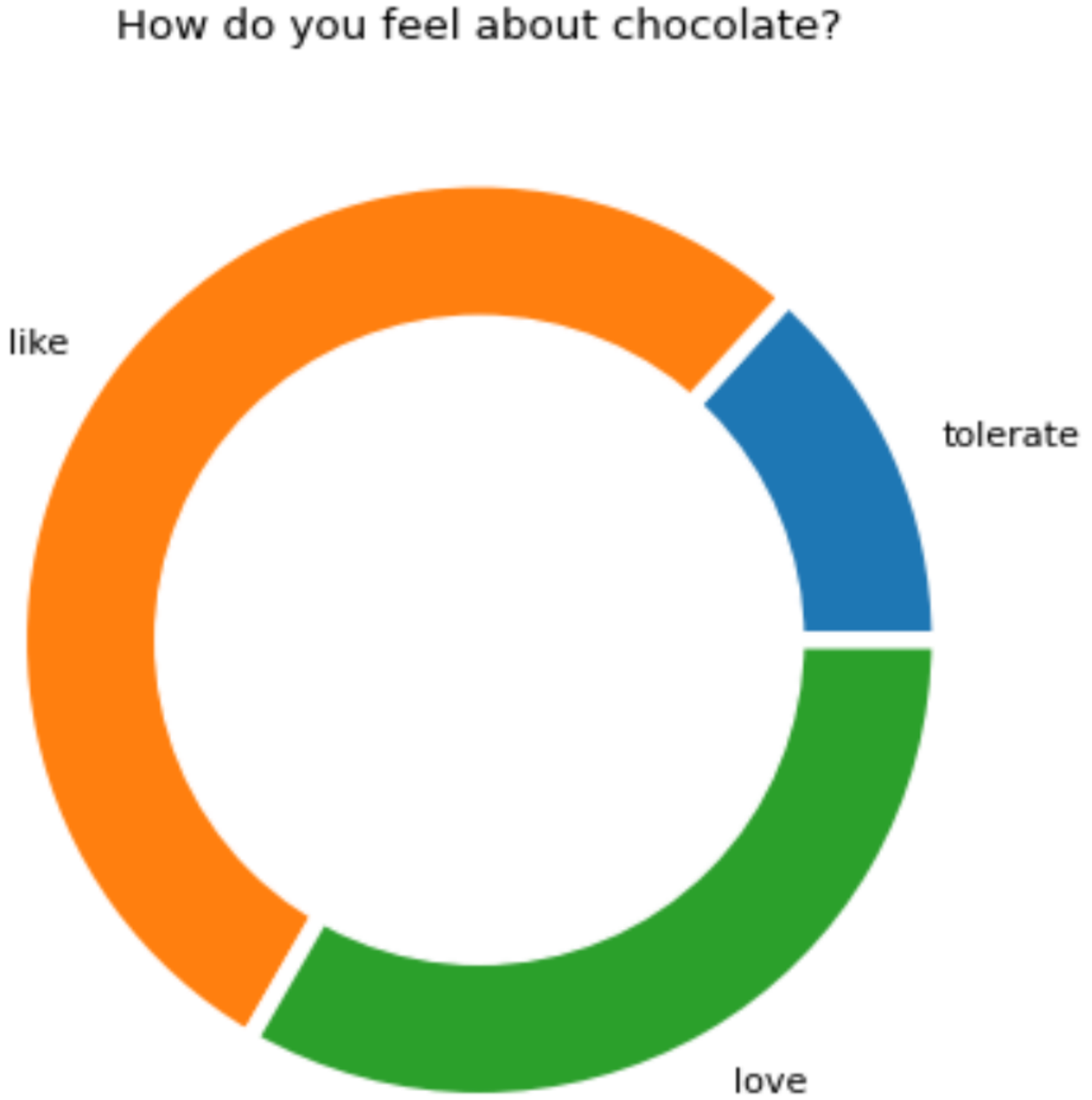
```
# Import your come from GitHub
# Use the magic %cd command to navigate around the file system
%cd /content/

# Use `isdir()` to see if the repository is already here
from genericpath import isdir

# get your code
if isdir("dgoodwin"):
  %cd dgoodwin
  print("let's pull the latest changes")
  !git pull
else:
  # Clone the repository with the latest code
  print("Nothing here, clone the repo")
  !git clone https://github.com/scrippscollege/DS_002.git dgoodwin

%cd /content/
```

Make a function to draw donut charts

```
def donutChart(keys=["yes","no"],vals=[33,20],label="Howdy!"):
  # Create a white circle at the center of the plot
  my_circle = plt.Circle( (0,0), 0.7, color='white')

  # Pie wedges with thick white edges
  props = {'linewidth':4, 'edgecolor':'white'}
  plt.pie(vals,labels=keys,wedgeprops=props )

  plt.title(label)
  p = plt.gcf()   # get current figure
  p.gca().add_artist(my_circle)
  # plt.show()

  return p
```

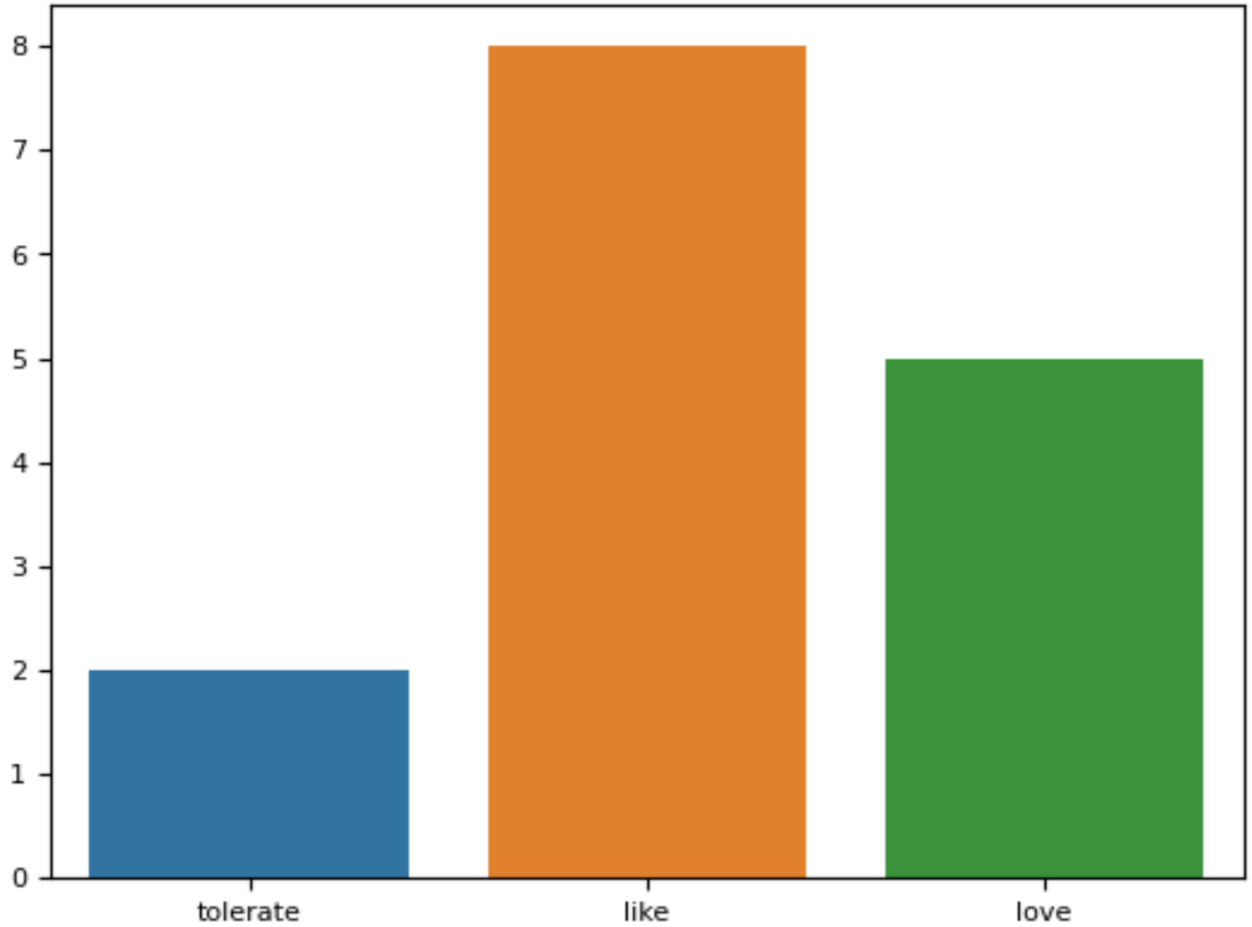
Preparing data for your function

```
# Turn that dictionary into two lists: one for keys, the other for values
keys = list(ord.keys())
vals = list(ord.values())

# make a donut
p = donutChart(keys=keys,vals=vals,label="How do you feel about chocolate?")
```



Seaborn's barplot

```
keys = list(ord.keys())
vals = list(ord.values())

sns.barplot(x=keys,y=vals)
plt.show()
```



Bonus: Opening a Google Sheet!

```
from google.colab import auth
auth.authenticate_user()
import gspread

# Authenticate with Google
from oauth2client.client import GoogleCredentials
gc = gspread.authorize(GoogleCredentials.get_application_default())

# Get the share link
sharelink = "https://docs.google.com/spreadsheets/d/1D-1pVbxWA-jVjckm0XJfWcA6g5EfBP07C9K9XsYOBHU/edit?usp=sharing"
wb = gc.open_by_url(sharelink)

# Get the right tab
sheet = wb.worksheet('Form Responses 1')
data = sheet.get_all_values()

# Exclude the header row
data[1:]
```

Howto: Homework

Answer these questions about the data:

1. **How many different hair colors are there in the class?**
2. **What is the most frequent response or median response about chocolate?**
3. **Could you calculate an average or mean response?**
4. **What is the minumum temperatures of our hometowns?**

get the data from Google Sheets

```
# A two dimensional list describing rows and columns
header = ["datetime","haircolor","chocolate","hometownTemp","hometownDistance"]

noirdata = data[1:]
```

Nominal data: Hair color?

```
from collections import Counter

# make a list of all the hair colors
allcolors = [row[1] for row in noirdata]

# Make your frequency counter
haircolor = Counter(allcolors)
haircolor
```

Ordinal data: chocolate preferences

```
# make a list of all the feelings about chocolate
allfeels = [row[2] for row in noirdata]

# Make your frequency counter
chocolate = Counter(allfeels)

# order the dictionary by value
chocList = sorted(chocolate.items(), key=lambda x: x[1], reverse=True)

allfeels,chocolate, chocList
```

Could you calculate an average or mean response for the chocolate data?

Ordinal data is ordered. We could try to coerce ordinals to intervals by giving each response a score. Take these bits from a popular book:

“With Likert scale data we cannot use the mean as a measure of central tendency as it has no meaning i.e. what is the average of Strongly agree and disagree?”

mean(), mode(), median(), median_grouped()

```
from statistics import mean, mode, median, median_grouped

scores = {
    'Hate':-2,
    'Tolerate':-1,
    'Neutral':0,
    'Like':1,
    'Love':2,
  }

print(f"allfeels contains {allfeels}")
print(f"The scores are given like this {scores}")

feelScores = []
for f in allfeels:
  myscore = scores[f]
  feelScores.append(myscore)

print(f"The feelScores list contains {feelScores}")

# Use these handy Python statistics functions to get the mean from the list
mean(feelScores), mode(feelScores), median(feelScores)

print(f"the most frequent vote was {mode(feelScores)}")
print(f"the mean score was {mean(feelScores)}")
print()
print(f"the median score was {median(feelScores)}")
print(f"The 50th percentile of data (median_grouped) is {median_grouped(feelScores)}")
```

Interval data: Average (mean) temperature

```
# Make a list of the current hometown temperatures
alltemps = [int(row[3]) for row in noirdata]

print(f"Current Hometown temperatures {alltemps}")
print()
print(f"the mean temperature was {mean(alltemps)}")
print(f"The 50th percentile of data (median_grouped) is {median_grouped(alltemps)}")
```

Warmest & coldest temps?

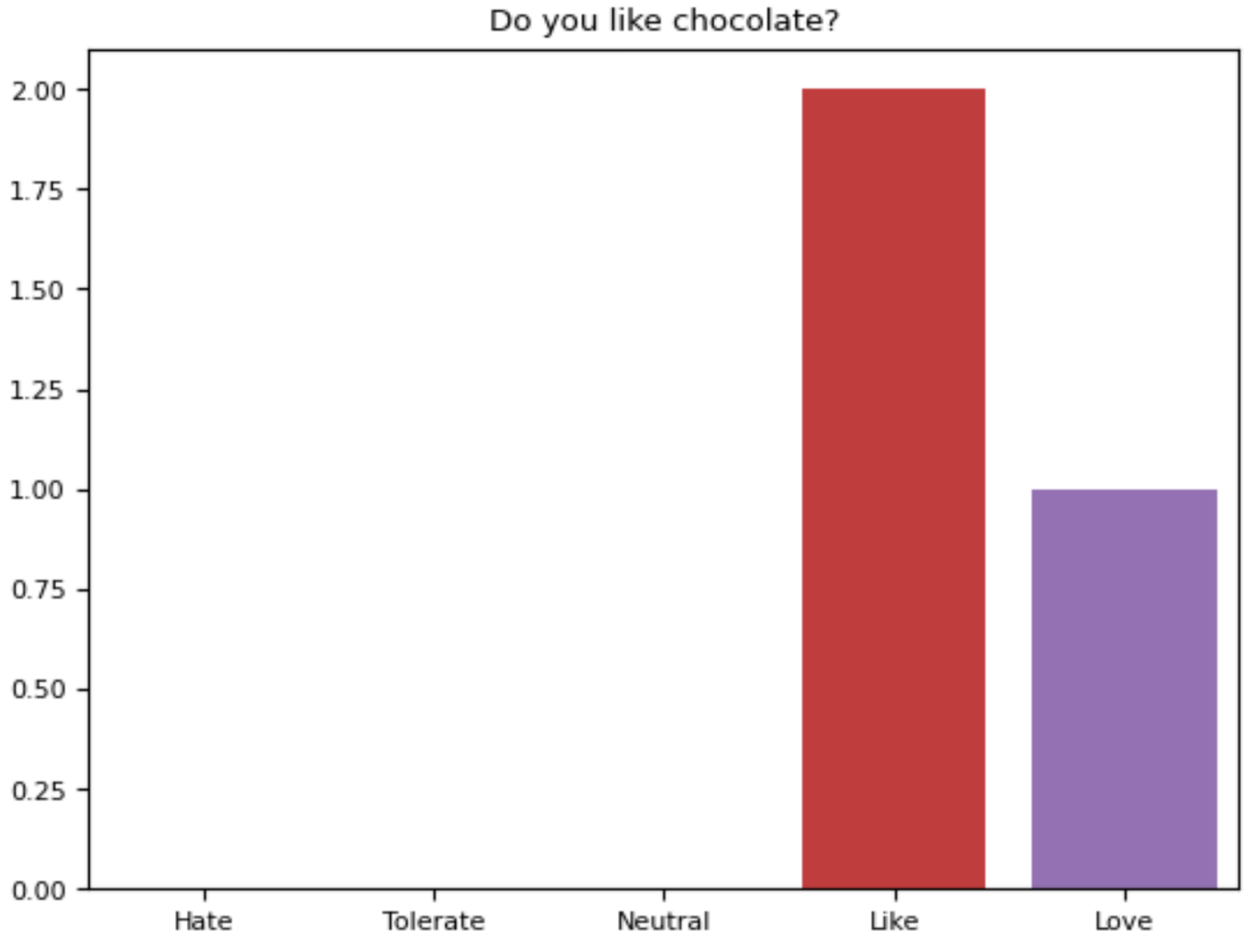
```
print(f"Current Hometown temperatures {alltemps}")
print()
print(f"the warmest temperature was {max(alltemps)}")
print()
print(f"Current Hometown temperatures {alltemps}")
print()
print(f"the coldest temperature was {min(alltemps)}")
```

Ratio data: distances to our hometowns

```
alldists = [int(row[4]) for row in noirdata]

print(f"All the distances {alldists}")
print()

print(f"the closest hometown is {min(alldists)} miles")
print(f"the furthest hometown is {max(alldists)} miles")
print(f"the mean distance is {mean(alldists)} miles")
print()
print(f"the median distance is {median(alldists)} miles")
print(f"The 50th percentile of data (median_grouped) is {median_grouped(alldists)} miles")
```

make an ordered barplot with zero values (if any)

```
from collections import OrderedDict

# Try a Seaborn barplot
keys = list(chocolate.keys())
vals = list(chocolate.values())

theSpread = OrderedDict([
              ('Hate',0),('Tolerate',0),('Neutral',0),('Like',0),('Love',0),
              ])

for k in chocolate.keys():
  theSpread[k] = chocolate[k]

# print(f"theSpread{theSpread}")

keys = list(theSpread.keys())
vals = list(theSpread.values())

sns.barplot(x=keys,y=vals).set(title='Do you like chocolate?')
plt.title = "hi"
plt.show()
```



Your turn! Homework

Copy this notebook and start answering questions!