The Diversity of First Digits

Diversity indices are used in many fields, including

- biology (to measure species diversity)
- economics (to measure the competition in an industry)
- political science (to measure the competitiveness in an election)

There are many different diversity indices in use, but they all involve first calculating the *distribution* of a categorical variable.

For example, suppose there are K species in an ecosystem. We record the species of n animals. The first step in calculating any diversity index is to determine the proportion of each species:

$$p_1, p_2, p_3, \ldots, p_K.$$

The diversity index depends on these proportions.

In this lab, you will learn about two diversity indices and apply them to the first digits that you looked at in Part A of this lab.

First, read in the S&P 500 data again (https://dlsun.github.io/pods/data/sp500.csv).

In [64]:
```
import pandas as pd
import numpy as np
from plotnine import *
```

In [65]:
```
df = pd.read_csv("https://dlsun.github.io/pods/data/sp500.csv")
df
```

Out[65]:

	date	Name	open	close	volume
0	2018-02-01	AAL	$54.00	$53.88	3623078
1	2018-02-01	AAPL	$167.16	$167.78	47230787
2	2018-02-01	AAP	$116.24	$117.29	760629
3	2018-02-01	ABBV	$112.24	$116.34	9943452
4	2018-02-01	ABC	$97.74	$99.29	2786798
...	...	...	...	...	...
500	2018-02-01	XYL	$72.50	$74.84	1817612
501	2018-02-01	YUM	$84.24	$83.98	1685275
502	2018-02-01	ZBH	$126.35	$128.19	1756300
503	2018-02-01	ZION	$53.79	$54.98	3542047
504	2018-02-01	ZTS	$76.84	$77.82	2982259

505 rows × 5 columns

In [66]:
```
df['first_digit'] = df['volume'].astype("str").str[0]
df['last_digit'] = df['volume'].astype("str").str[-1]
df
```

Out[66]:

	date	Name	open	close	volume	first_digit	last_digit
0	2018-02-01	AAL	$54.00	$53.88	3623078	3	8
1	2018-02-01	AAPL	$167.16	$167.78	47230787	4	7
2	2018-02-01	AAP	$116.24	$117.29	760629	7	9
3	2018-02-01	ABBV	$112.24	$116.34	9943452	9	2
4	2018-02-01	ABC	$97.74	$99.29	2786798	2	8
...	...	...	...	...	...	...	...
500	2018-02-01	XYL	$72.50	$74.84	1817612	1	2
501	2018-02-01	YUM	$84.24	$83.98	1685275	1	5
502	2018-02-01	ZBH	$126.35	$128.19	1756300	1	0
503	2018-02-01	ZION	$53.79	$54.98	3542047	3	7
504	2018-02-01	ZTS	$76.84	$77.82	2982259	2	9

505 rows × 7 columns

Question 1: Simpson Diversity Index

The Simpson diversity index (also called the Herfindahl–Hirschman index in economics) is the probability that two randomly selected observations belong to the same category. The higher the probability, the lower the diversity.

The formula for the Simpson diversity index is

$$\sum_{i=1}^{K} p_i^2.$$

Connections to Other Courses:

- In STATS 117, you will learn the Law of Total Probability, which is used to show that this is in fact the probability that two randomly selected observations belong to the same category.
- In MATH 51, you will learn Lagrange Multipliers, which is used to show that the minimum possible value of this index (maximum diversity) is achieved when every category has the same proportion $p_i = 1/K$.

Calculate the Simpson diversity indices for

1. the first digit of the **volume** in the S&P data
2. the last digit of the **volume** in the S&P data

How do the diversities compare?

In [67]:
```
#Calculating the Simpson Diversity Index for the first digits
first_simpson = (df['first_digit'].value_counts(normalize=True) ** 2).sum()
first_simpson
```

Out[67]:
```
0.1793196745417116
```

In [68]:
```
#Calculating the Simpson Diversity Index for the last digits
last_simpson = (df['last_digit'].value_counts(normalize=True) ** 2).sum()
last_simpson
```

Out[68]:
```
0.10092736006273895
```

YOUR INTERPRETATION HERE

The Simpson Diversity index is lower for the last digits in each of the volume values, with approximately 0.1 for the last digits and about 0.179 for the first digits. The relation between the two diversity indeces indicates that there is a greater probability of two first digits belonging to the same category (two different volume values having the same first digit) as opposed to the last digits. This is most likely because the frequencies of the first digits follow a linear pattern (the frequency of occurrance as a first digit decreases as the value of the digit increases) as opposed to the distribution of last digits, which is nearly even and therefore leaves probability to random chance. Thus, the distribution of the last digits is more diverse, as the probability that any two randomly selected last digits are the same is less and there is a nearly equal chance of each fitting into any given category.

Question 2: Shannon Entropy

The Shannon entropy measures the difficulty of predicting the category of a randomly selected observation. The more difficult the prediction, the higher the diversity.

The formula for the Shannon entropy is

$$-\sum_{i=1}^{K} p_i \log p_i.$$

Calculate the Shannon entropy for

1. the first digit of the **volume** in the S&P data
2. the last digit of the **volume** in the S&P data

How do the diversities compare?

Hint: If you want a vectorized form of the $\log$ function, you can use the $\log$ function in Numpy:

```
import numpy as np
np.log()
```

In [69]:
```
#Calculating the Shannon Entropy for the first digits
first_shannon =  -1 * (df['first_digit'].value_counts(normalize=True) * np.log
first_shannon
```

Out[69]:
```
1.9473465331779622
```

In [70]:
```
#Calculating the Shannon Entropy for the last digits
last_shannon =  -1 * (df['last_digit'].value_counts(normalize=True) * np.log(d
last_shannon
```

Out[70]:
```
2.2977862376301905
```

YOUR INTERPRETATION HERE

The Shannon Entropy values (approximately 1.95 for first digits and about 2.3 for last digits) demonstrate that the last digits in the values are more diverse and it is therefore more difficult to predict the last digit of any randomly volume value than it is to predict the first digit. This is because the distribution of the last digits for the volume values in the S&P 500 dataset is much more even across the possible digits and does not follow any particular pattern as the first digits do. Since each digit has a nearly equal chance of appearing as the last digit in the volume values, there is no definitive way to predict the category of any given digit 0-9. This supports the conclusion drawn from calculating the Simpson Diversity Indeces as well because both of these values indicated that the distribution of the last digits was more diverse.

Question 3: Another Data Set

The first-digit distribution that you discovered in Part A of this lab is called Benford's Law. It holds for many data sets, but not all data sets. Here's a video about how some people erroneously applied Benford's Law to the 2020 election results.

In this question, you will find a quantitative variable from a different data set, and compare its first-digit distribution to the S&P data. You should find another data set yourself; if you don't know where to look, this is a good place to start.

Make sure your quantitative variable satisfies the following conditions:

- All values are positive. (They cannot be equal to 0.)
- The values have varying numbers of digits.

Upload your data set to Colab by clicking on the folder icon at left. If the file you uploaded was called `my_data.csv`, you can read in this data by doing `pd.read_csv("my_data.csv")`.

How does its distribution of first digits compare to the S&P **volume**s? Does it follow Benford's Law?

PROVIDE SOME CONTEXT FOR THE DATA SET YOU CHOSE HERE

This Breaches dataset displays the information for various cybersecurity data breaches that occurred in business and hospitals across the U.S. Territories between 2009 and 2014. This information includes the various organizations and businesses involved, the date of the breach, the type of security breach, the device that was breached, the date of the breach, and the number of individuals who were affected. For this exerise, I will be taking a look at the first digits in the number of individuals affected for each breach and see if the distribution follows Benford's Law.

In [71]:
```
df = pd.read_csv("/content/breaches.csv")
df
```

Out[71]:

	rownames	Number	Name_of_Covered_Entity	State	Business_Associate_Involved	Indivi
0	1	0	Brooke Army Medical Center	TX	NaN	
1	2	1	Mid America Kidney Stone Association, LLC	MO	NaN	
2	3	2	Alaska Department of Health and Social Services	AK	NaN	
3	4	3	Health Services for Children with Special Need...	DC	NaN	
4	5	4	L. Douglas Carlson, M.D.	CA	NaN	
...	...	...	...	...	...	
1050	1051	1050	Puerto Rico Health Insurance	PR	American Health Inc	
1051	1052	1051	Hospitalists of Brandon, LLC	FL	Doctors First Choice Billings, Inc.	
1052	1053	1052	Santa Rosa Memorial Hospital	CA	NaN	
1053	1054	1053	Group Health Plan of Hurley Medical Center	MI	NaN	
1054	1055	1054	Abrham Tekola, M.D.,INC	CA	NaN	

1055 rows × 14 columns

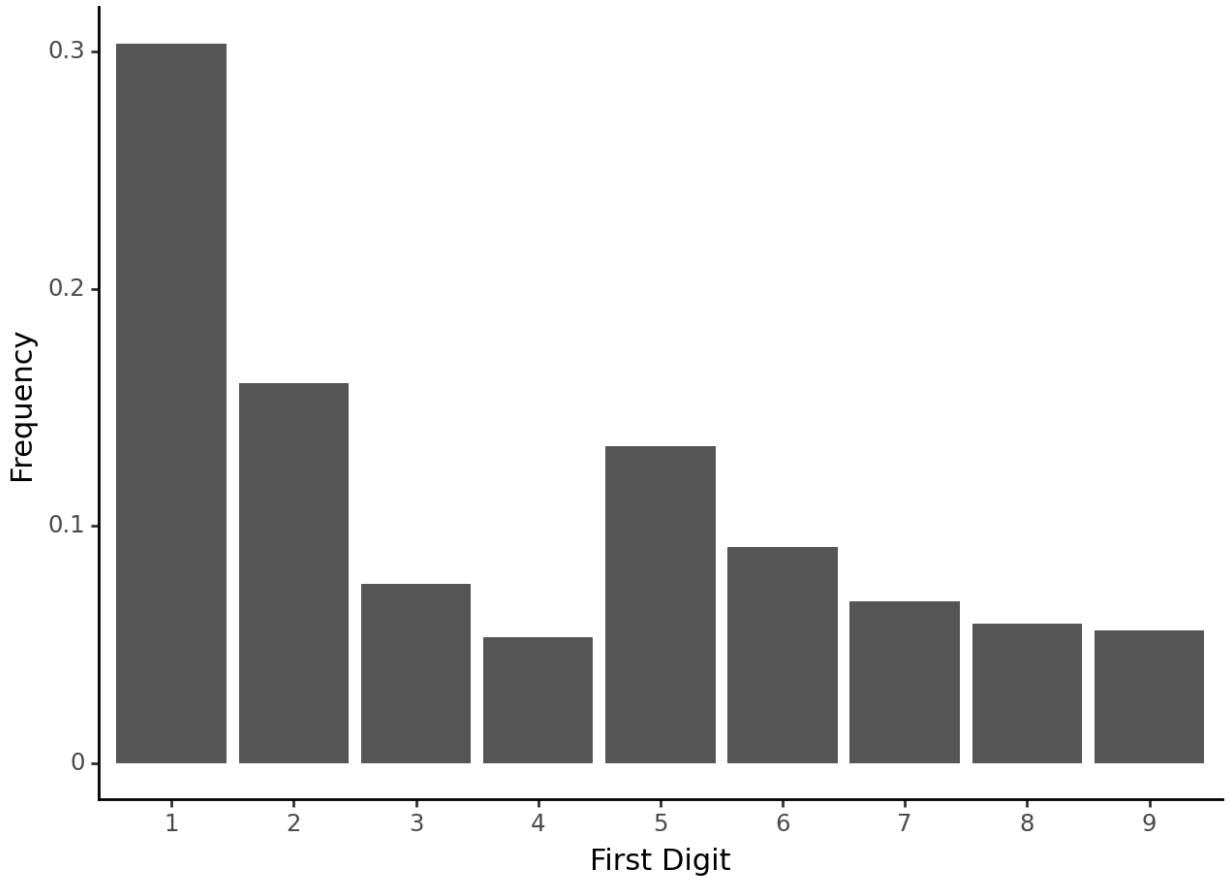
In [72]:
```
df['first_int'] = df['Individuals_Affected'].astype("str").str[0]
first_dist = df["first_int"].value_counts(normalize=True).to_frame().reset_inde
first_dist
```

Out[72]:

	first_int	proportion
0	1	0.303318
1	2	0.160190
2	5	0.133649
3	6	0.090995
4	3	0.075829
5	7	0.068246
6	8	0.058768
7	9	0.055924
8	4	0.053081

In [73]:
```
first_dist["proportion"] = first_dist["proportion"].astype("float")
```

In [74]:
```
(ggplot(first_dist, aes(x = "first_int", y = "proportion"))
+ geom_col()
+ xlab("First Digit")
+ ylab("Frequency")
+ theme_classic()
)
```



Out[74]: <Figure Size: (640 x 480)>

YOUR INTERPRETATION HERE

The distribution shown above of the first digits in each number of individuals affected in the breaches differs from that of the first digits in the volume values of the S&P 500 dataset in that it is not as linear. While 1 is still the leading first digit and appears as the first digit over 30% of the time, there is not a fully consistent decrease in the frequency of each digit occuring as the value of the digit itself increases. This can be seen in that the 5 occurs more frequently (about 15% of the time) than the rest of the digits excluding 1, breaking the directly linear and proportional relationship seen in the S&P 500 data. All in all, the distributions of first digits in both datasets follow Benford's law, though the volume values in the S&P 500 data demonstrated a more linear relationship.

Question 4: Comparing Diversities

How does the diversity of the first digits in your variable compare to the diversity of the first digits in the S&P **volume**s?

In [ ]:
```
#Calculating the Simpson Diversity Index for the first digits
first_simpson_stars = (df['first_int'].value_counts(normalize=True) ** 2).sum(
first_simpson_stars
```

Out[ ]:
```
0.16361088025875428
```

In [ ]:
```
#Calculating the Shannon Entropy for the first digits in the stars dataset
first_shannon_stars =  -1 * (df['first_int'].value_counts(normalize=True) * np
first_shannon_stars
```

Out[ ]:
```
2.004777562983429
```

YOUR INTERPRETATION HERE

Both the Simpson Diversity index and the Shannon Entropy for the first digits in the breaches data were approximately 0.2 lower than those for the S&P 500 dataset (about 0.16 versus about 0.179 for Simpson Diveristy index and 2.0 versus about 2.3 for the Shannon Entropies). The former indicates that there was a slightly higher probability of predicting whether the first digits in two randomly selected volume values (S&P 500 dataset) were the same as opposed to the number of individuals affected by breaches. This is most likely because while the distribution of first digits in the S&P 500 dataset follows a direct pattern, the frequencies of digits 2-9 being first digits in the breaches dataset are not as linear and therefore less predictable.

The Shannon Entropy for the first digits in the breaches data was also slightly lower than that in the S&P 500 data, indicating that the category of the first digits in breaches data was more difficult to predict. This is also most likley because the first digits in the breaches dataset do not follow directly linear and proportional relationship the same way the first digits in the S&P 500 dataset do. This makes it more difficult to distinguish a clear pattern and make an educated prediction of what the first digit will be.

Submission Instructions

- Restart this notebook and run the cells from beginning to end.
  - Go to Runtime > Restart and Run All.

In [ ]:
```
# @markdown Run this cell to download this notebook as a webpage, `_NOTEBOOK.h

import google, json, nbformat

# Get the current notebook and write it to _NOTEBOOK.ipynb
raw_notebook = google.colab._message.blocking_request("get_ipynb",
                                                      timeout_sec=30)["ipynb"]
with open("_NOTEBOOK.ipynb", "w", encoding="utf-8") as ipynb_file:
  ipynb_file.write(json.dumps(raw_notebook))

# Use nbconvert to convert .ipynb to .html.
!jupyter nbconvert --to html --log-level WARN _NOTEBOOK.ipynb

# Download the .html file.
google.colab.files.download("_NOTEBOOK.html")
```

- Open `_NOTEBOOK.html` in your browser, and save it as a PDF.
  - Go to File > Print > Save as PDF.
- Double check that all of your code and output is visible in the saved PDF.
- Upload the PDF to Gradescope.
  - Please be sure to select the correct pages corresponding to each question.