1B. Distribution of First Digits

January 16, 2019

1 The Distribution of First Digits

In this lab, you will explore the distribution of first digits in real data. For example, the first digits of the numbers 52, 30.8, and 0.07 are 5, 3, and 7 respectively. In this lab, you will investigate the question: how frequently does each digit 1-9 appear as the first digit of the number?

1.1 Question 0

Make a prediction.

1. Approximately what percentage of the values do you think will have a *first* digit of 1? What percentage of the values do you think will have a first digit of 9?
2. Approximately what percentage of the values do you think will have a *last* digit of 1? What percentage of the values do you think will have a last digit of 9?

(Don't worry about being wrong. You will earn full credit for any justified answer.)

ENTER YOUR WRITTEN EXPLANATION HERE. After taking STAT 305, the probability of digits should be equally likely, so I predict 11.11% of values will have a first digit of 1 and 11.11% will have a first digit of 9. Also for the same reason, I predict 11.11% of values will have a last digit of 1 and 11.11% of values will have a last digit of 9.

1.2 Question 1

The S&P 500 is a stock index based on the market capitalizations of large companies that are publicly traded on the NYSE or NASDAQ. The CSV file `https://raw.githubusercontent.com/dlsun/data-science-book/master/data/sp500.csv` contains data from February 1, 2018 about the stocks that comprise the S&P 500. We will investigate the first digit distributions of the variables in this data set.

Read in the S&P 500 data. What is the unit of observation in this data set? Is there a variable that is natural to use as the index? If so, set that variable to be the index. Once you are done, display the `DataFrame`.

```
In [1]: %matplotlib inline
        import pandas as pd
        pd.options.display.max_rows = 10

        df = pd.read_csv("https://raw.githubusercontent.com/dlsun/data-science-book/master/data
```

```
df = df.set_index("Name")
df
```

Out[1]:

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

[505 rows x 4 columns]

ENTER YOUR WRITTEN EXPLANATION HERE. The unit of observation is the volume for this dataset. The name variable is natural to use as the index since all the names are unique.

1.3 Question 2

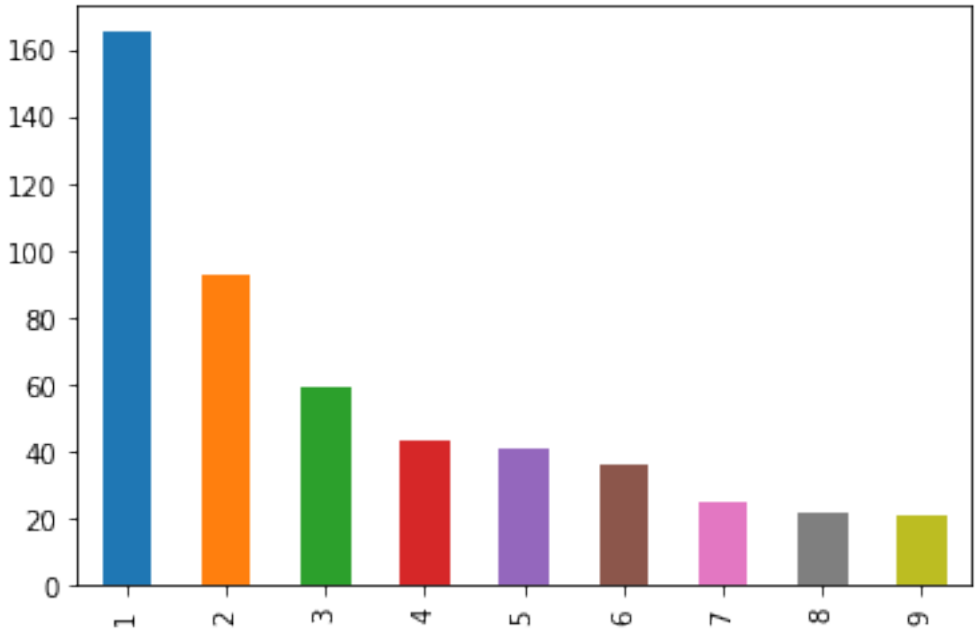
We will start by looking at the `volume` column. This variable tells us how many shares were traded on that date.

Extract the first digit of every value in this column. (*Hint:* First, turn the numbers into strings. Then, use the text processing functionalities of `pandas` to extract the first character of each string.) Make an appropriate visualization to display the distribution of the first digits. (*Hint:* Think carefully about whether the variable you are plotting is quantitative or categorical.)

How does this compare with what you predicted in Question 0?

In [2]:
```
df["volume"] = df["volume"].astype(str)
volume_as_series = pd.Series(df["volume"])
df["first"] = volume_as_series.str[0]
df["first"].value_counts().plot.bar()
```

Out[2]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1adb730128>
```



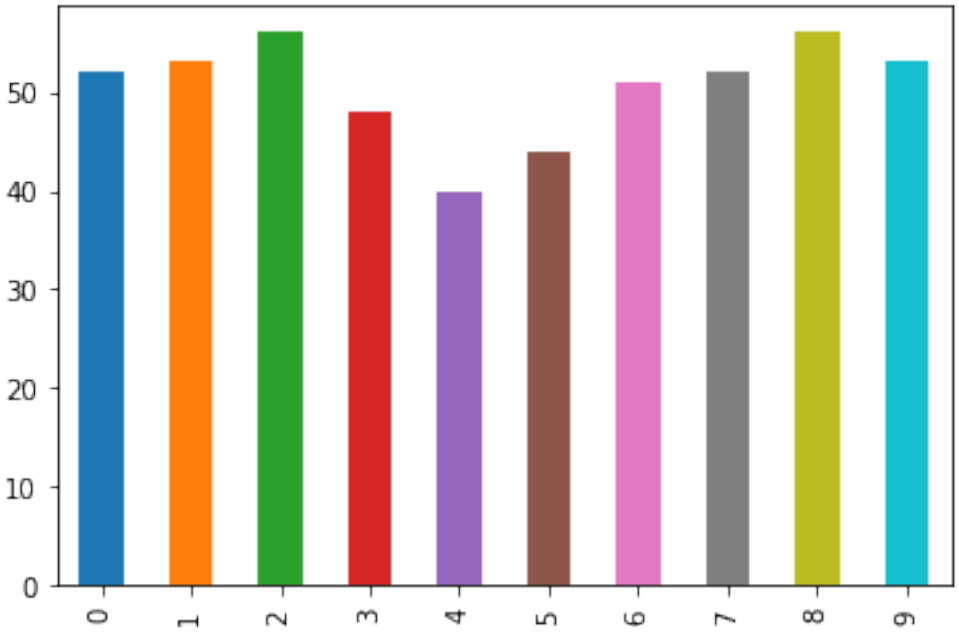
ENTER YOUR WRITTEN EXPLANATION HERE. The distribution of the first digit of volume compares quite differently to what I predicted in Question 0. I had hoped for a uniform distribution, but instead we see a skewed right distribution of values from least to greatest, making '1' the most common first value of volume of stocks in S&P500 and '9' the least common first value of volume of stocks in S&P500. Therefore, the probability of each value varies and overall, the values 1-9 are not equally likely.

1.4 Question 3

Now, repeat Question 2, but for the distribution of *last* digits. Again, make an appropriate visualization and compare with your prediction in Question 0.

```
In [3]: df["last"] = volume_as_series.str[-1]
        last_count = df["last"].value_counts()
        last_count.sort_index(inplace=True)
        last_count.plot.bar()

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1ad9602710>
```



ENTER YOUR WRITTEN EXPLANATION HERE. In Question 0, I predicted the distribution of values for the last digit would be equally likely and it appears from this distribution, that each of the values are almost equally likely. There is some slight differences in probabilites but overall there is no major skewness or favoritism to any particular value since the distribution of last digits is almost uniform.

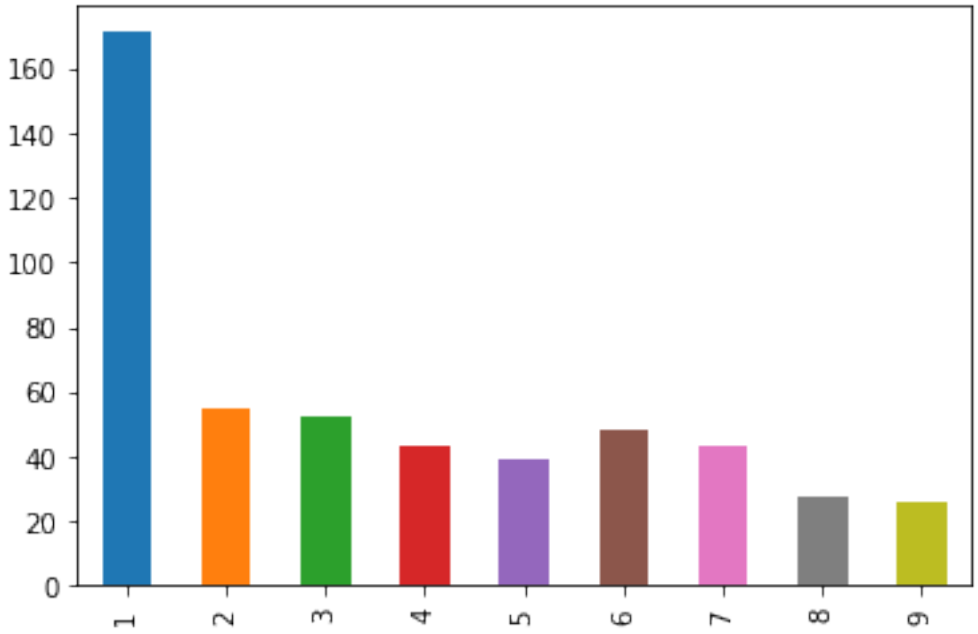
1.5 Question 4

Maybe the `volume` column was just a fluke. Let's see if the first digit distribution holds up when we look at a very different variable: the closing price of the stock. Make a visualization of the first digit distribution of the closing price (the `close` column of the `DataFrame`). Comment on what you see.

(*Hint:* What type did `pandas` infer this variable as and why? You will have to first clean the values using the text processing functionalities of `pandas` and then convert this variable to a quantitative variable.)

```
In [4]: closing_price_as_series = pd.Series(df["close"])
        df["close first"] = closing_price_as_series.str[1]
        close_first_counts = df["close first"].value_counts()
        close_first_counts.sort_index(inplace=True)
        close_first_counts.plot.bar()

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1ad957b6a0>
```



ENTER YOUR WRITTEN EXPLANATION HERE. Pandas infers the "closing" variable to be an object because of the dollar sign symbol. From the distribution of first values of S&P500 closing price data, we see that digits 1-9 are not equally likely. Once again, the values are skewed right with the number "one" again being most common to any other number.

1.6 Submission Instructions

Once you are finished, follow these steps:

1. Restart the kernel and re-run this notebook from beginning to end by going to `Kernel > Restart Kernel and Run All Cells`.
2. If this process stops halfway through, that means there was an error. Correct the error and repeat Step 1 until the notebook runs from beginning to end.
3. Double check that there is a number next to each code cell and that these numbers are in order.

Then, submit your lab as follows:

1. Go to `File > Export Notebook As > PDF`.
2. Double check that the entire notebook, from beginning to end, is in this PDF file. (If the notebook is cut off, try first exporting the notebook to HTML and printing to PDF.)
3. Upload the PDF to PolyLearn.