Homework due January 28 - Analyzing and Graphing Data

There are 2 questions in this assignment.

Question 1: Superbowl Final Scores for Pool

In class, we performed the necessary data analysis to see the distribution of last digits for the combined half time scores. Use similar methods appplied to the pandas dataframe to examine the frequency distribution for last digits of the combined final scores.

The data is available at:
url2="https://bulldog2.redlands.edu/fac/tamara_veenstra/ML/superbowl_stats.csv"

Repeat the work we did to clean the data, add the necessary columns to the data frame for the combined final scores, create a frequency count (in Pandas) and corresponding bar chart (in Matplotlib). Which numbers are best and worst? Are you surprised?

You should include commented code (with your own comments!) for the new work, and a conclusion discussing which numbers are best/worst and if that was surprising.

```
# import packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
# import data
url2="https://bulldog2.redlands.edu/fac/tamara_veenstra/ML/superbowl_stats.csv"
df=pd.read_csv(url2)
```

```
# checked to make sure everything was imported properly
df
```

2	2018	Eagles (NFC, 13-3)	9.0	13.0	7.0	12.0	41.0	England Patriots (AFC, 13-3)	3.0	9.0	14.0	
3	2017	New England Patriots (AFC, 14-2)	0.0	3.0	6.0	25.0	34.0	Atlanta Falcons (NFC, 11-5)	0.0	21.0	7.0	
4	2016	Carolina Panthers (NFC, 15-1)	0.0	7.0	0.0	3.0	10.0	Denver Broncos (AFC, 12-4)	10.0	3.0	3.0	
5	2015	New England Patriots (AFC, 12-4)	0.0	14.0	0.0	14.0	28.0	Seattle Seahawks (NFC, 12-4)	0.0	14.0	10.0	
6	2014	Seattle Seahawks (NFC, 13-3)	8.0	14.0	14.0	7.0	43.0	Denver Broncos (AFC, 13-3)	0.0	0.0	8.0	
7	2013	Baltimore Ravens (AFC, 10-6)	7.0	14.0	7.0	6.0	34.0	San Francisco 49ers (NFC, 11-4-1)	3.0	3.0	17.0	
8	2012	New York Giants (NFC, 9-7)	9.0	0.0	6.0	6.0	21.0	New England Patriots (AFC, 13-3)	0.0	10.0	7.0	
9	2011	Pittsburgh Steelers (AFC, 12-4)	0.0	10.0	7.0	8.0	25.0	Green Bay Packers (NFC, 10-6)	14.0	7.0	0.0	1
10	2010	New Orleans Saints (NFC, 13-3)	0.0	6.0	10.0	15.0	31.0	Indianapolis Colts (AFC, 14-2)	10.0	0.0	7.0	
11	2009	Pittsburgh Steelers (AFC, 12-4)	3.0	14.0	3.0	7.0	27.0	Arizona Cardinals (NFC, 9-7)	0.0	7.0	0.0	1
12	2008	New York Giants (NFC, 10-6)	3.0	0.0	0.0	14.0	17.0	New England Patriots (AFC, 16-0)	0.0	7.0	0.0	
13	2007	Indianapolis Colts (AFC, 12-4)	6.0	10.0	6.0	7.0	29.0	Chicago Bears (NFC, 13-3)	14.0	0.0	3.0	
14	2006	Seattle Seahawks (NFC, 13-3)	3.0	0.0	7.0	0.0	10.0	Pittsburgh Steelers (AFC, 11-5)	0.0	7.0	7.0	
15	2005	New England	0.0	7.0	7.0	10.0	24.0	Philadelphia Eagles	0.0	7.0	7.0	

		Patriots (AFC, 14-2)					(NFC, 13-3)					
16	2004	Carolina Panthers (NFC, 11-5)	0.0	10.0	0.0	19.0	29.0	New England Patriots (AFC, 14-2)	0.0	14.0	0.0	1
17	2003	Oakland Raiders (AFC, 11-5)	3.0	0.0	6.0	12.0	21.0	Tampa Bay Buccaneers (NFC, 12-4)	3.0	17.0	14.0	1
18	2002	St. Louis Rams (NFC, 14-2)	3.0	0.0	0.0	14.0	17.0	New England Patriots (AFC, 11-5)	0.0	14.0	3.0	
19	2001	Baltimore Ravens (AFC, 12-4)	7.0	3.0	14.0	10.0	34.0	New York Giants (NFC, 12-4)	0.0	0.0	7.0	
20	2000	St. Louis Rams (NFC, 13-3)	3.0	6.0	7.0	7.0	23.0	Tennessee Titans (AFC, 13-3)	0.0	0.0	6.0	1
21	1999	Denver Broncos (AFC, 14-2)	7.0	10.0	0.0	17.0	34.0	Atlanta Falcons (NFC, 14-2)	3.0	3.0	0.0	1
22	1998	Green Bay Packers (NFC, 13-3)	7.0	7.0	3.0	7.0	24.0	Denver Broncos (AFC, 12-4)	7.0	10.0	7.0	
23	1997	New England Patriots (AFC, 11-5)	14.0	0.0	7.0	0.0	21.0	Green Bay Packers (NFC, 13-3)	10.0	17.0	8.0	
24	1996	Dallas Cowboys (NFC, 12-4)	10.0	3.0	7.0	7.0	27.0	Pittsburgh Steelers (AFC, 11-5)	0.0	7.0	0.0	1
25	1995	San Diego Chargers (AFC, 11-5)	7.0	3.0	8.0	8.0	26.0	San Francisco 49ers (NFC, 13-3)	14.0	14.0	14.0	
26	1994	Dallas Cowboys (NFC, 12-4)	6.0	0.0	14.0	10.0	30.0	Buffalo Bills (AFC, 12-4)	3.0	10.0	0.0	
27	1993	Buffalo Bills (AFC, 11-5)	7.0	3.0	7.0	0.0	17.0	Dallas Cowboys (NFC, 13-3)	14.0	14.0	3.0	2

```
# clean up the code
df.drop(columns="Unnamed: 13", inplace=True)
df.drop([54,55,57], inplace=True)
df.dropna(inplace=True)
df.tail()
```

	Year	Team 1	1st	2nd	3rd	4th	Final	Team 2	1st.1	2nd.1	3rd.1
49	1971	Baltimore Colts (AFC, 11-2-1)	0.0	6.0	0.0	10.0	16.0	Dallas Cowboys (NFC, 10-4)	3.0	10.0	0.0
50	1970	Minnesota Vikings (NFL, 12-2)	0.0	0.0	7.0	0.0	7.0	Kansas City Chiefs (AFL, 11-3)	3.0	13.0	7.0
51	1969	New York Jets	0.0	7.0	6.0	3.0	16.0	Baltimore Colts (NFL	0.0	0.0	0.0

```
# viewing the dataframe to make sure it all looks good
df
```

	Year	Team 1	1st	2nd	3rd	4th	Final	Team 2	1st.1	2nd.1	3rd.1
0	2020	San Francisco 49ers (NFC, 13-3)	3.0	7.0	10.0	0.0	20.0	Kansas City Chiefs (AFC, 12-4)	7.0	3.0	0.0
1	2019	New England Patriots (AFC, 11-5)	0.0	3.0	0.0	10.0	13.0	Los Angeles Rams (NFC, 13-3)	0.0	0.0	3.0
2	2018	Philadelphia Eagles (NFC, 13-3)	9.0	13.0	7.0	12.0	41.0	New England Patriots (AFC, 13-3)	3.0	9.0	14.0
3	2017	New England Patriots (AFC, 14-2)	0.0	3.0	6.0	25.0	34.0	Atlanta Falcons (NFC, 11-5)	0.0	21.0	7.0
4	2016	Carolina Panthers (NFC, 15-1)	0.0	7.0	0.0	3.0	10.0	Denver Broncos (AFC, 12-4)	10.0	3.0	3.0
5	2015	New England Patriots (AFC, 12-4)	0.0	14.0	0.0	14.0	28.0	Seattle Seahawks (NFC, 12-4)	0.0	14.0	10.0
6	2014	Seattle Seahawks (NFC, 13-3)	8.0	14.0	14.0	7.0	43.0	Denver Broncos (AFC, 13-3)	0.0	0.0	8.0
7	2013	Baltimore Ravens (AFC, 10-6)	7.0	14.0	7.0	6.0	34.0	San Francisco 49ers (NFC, 11-4-1)	3.0	3.0	17.0
8	2012	New York Giants (NFC, 9-7)	9.0	0.0	6.0	6.0	21.0	New England Patriots (AFC, 13-3)	0.0	10.0	7.0
9	2011	Pittsburgh Steelers (AFC, 12-4)	0.0	10.0	7.0	8.0	25.0	Green Bay Packers (NFC, 10-6)	14.0	7.0	0.0
10	2010	New Orleans Saints (NFC, 13-3)	0.0	6.0	10.0	15.0	31.0	Indianapolis Colts (AFC, 14-2)	10.0	0.0	7.0
11	2009	Pittsburgh Steelers (AFC, 12-4)	3.0	14.0	3.0	7.0	27.0	Arizona Cardinals (NFC, 9-7)	0.0	7.0	0.0
12	2008	New York Giants (NFC, 10-6)	3.0	0.0	0.0	14.0	17.0	New England Patriots (AFC, 16-0)	0.0	7.0	0.0
13	2007	Indianapolis Colts (AFC, 12-4)	6.0	10.0	6.0	7.0	29.0	Chicago Bears (NFC, 13-3)	14.0	0.0	3.0
14	2006	Seattle Seahawks	3.0	0.0	7.0	0.0	10.0	Pittsburgh Steelers	0.0	7.0	7.0

14	2006	Seahawks (NFC, 13-3)	3.0	0.0	7.0	0.0	10.0	Steelers (AFC, 11-5)	0.0	7.0	7.0
15	2005	New England Patriots (AFC	0.0	7.0	7.0	10.0	24.0	Philadelphia Eagles (NFC	0.0	7.0	7.0

```
# checking the datatypes
df.dtypes
```

```
Year        object
Team 1      object
1st        float64
2nd        float64
3rd        float64
4th        float64
Final      float64
Team 2      object
1st.1      float64
2nd.1      float64
3rd.1      float64
4th.1      float64
Final.1    float64
dtype: object
```

20	2000	Rams (NFC	3.0	6.0	7.0	7.0	23.0	Titans (AFC	0.0	0.0	6.0

```
# changing the datatype of year to be an int
df["Year"]=pd.to_numeric(df["Year"],errors='coerce',downcast='integer')
```

21	1999	Broncos (AFC,	7.0	10.0	0.0	17.0	34.0	Falcons	3.0	3.0	0.0

```
# confirming that the datatype of year was, in fact, changed
df.dtypes
```

```
Year         int16
Team 1      object
1st        float64
2nd        float64
3rd        float64
4th        float64
Final      float64
Team 2      object
1st.1      float64
2nd.1      float64
3rd.1      float64
4th.1      float64
Final.1    float64
dtype: object
```

```
# setting an index
df.set_index('Year', inplace=True)
df.head()
```

Year	Team 1	1st	2nd	3rd	4th	Final	Team 2	1st.1	2nd.1	3rd.1
2020	San Francisco 49ers (NFC, 13-3)	3.0	7.0	10.0	0.0	20.0	Kansas City Chiefs (AFC, 12-4)	7.0	3.0	0.0

```
# changing the names in the columns
df.rename(columns={'1st':'Team1 1st', '2nd':'Team1 2nd', '3rd':'Team1 3rd', '4th':'Team1 4th'
for col in df.columns:  ##loop through all the column names
    if col[-2:]==".1":  ## check if the end with .1
        new_name="Team2 "+col[0:-2] ## create a new name which is Team2 concatenated with the
        new_name_dict={col:new_name} ## dictionary has a single element in the form {old_name
        df.rename(columns={col:new_name}, inplace=True) ## use rename for each column
df.head()
```

Year	Team 1	Team1 1st	Team1 2nd	Team1 3rd	Team1 4th	Team1 Final	Team 2	Team2 1st	Team2 2nd	Team2 3rd
2020	San Francisco 49ers (NFC, 13-3)	3.0	7.0	10.0	0.0	20.0	Kansas City Chiefs (AFC, 12-4)	7.0	3.0	0.0
2019	New England Patriots (AFC, 11-5)	0.0	3.0	0.0	10.0	13.0	Los Angeles Rams (NFC, 13-3)	0.0	0.0	3.0
2018	Philadelphia Eagles (NFC, 13-3)	9.0	13.0	7.0	12.0	41.0	New England Patriots (AFC, 13-3)	3.0	9.0	14.0
2017	New England Patriots (AFC,	0.0	3.0	6.0	25.0	34.0	Atlanta Falcons	0.0	21.0	7.0

```
# adding column for the final combined score
df['Combined final']=df['Team1 Final']+df['Team2 Final']
df.head()
```

	Team 1	Team1 1st	Team1 2nd	Team1 3rd	Team1 4th	Team1 Final	Team 2	Team2 1st	Team2 2nd	Team2 3rd	Team2 4th
Year											

```
# showing just the last digit of the final score
df["Final Pool"]=df['Combined final']%10
df.head()
```

	Team 1	Team1 1st	Team1 2nd	Team1 3rd	Team1 4th	Team1 Final	Team 2	Team2 1st	Team2 2nd	Team2 3rd	Team2 4th
Year											
2020	San Francisco 49ers (NFC, 13-3)	3.0	7.0	10.0	0.0	20.0	Kansas City Chiefs (AFC, 12-4)	7.0	3.0	0.0	21.0
2019	New England Patriots (AFC, 11-5)	0.0	3.0	0.0	10.0	13.0	Los Angeles Rams (NFC, 13-3)	0.0	0.0	3.0	0.0
	Philadelphia						New England				

```
# get a frequency count
result=df['Final Pool'].value_counts()
result
```

```
1.0    9
6.0    8
7.0    7
9.0    6
5.0    6
4.0    5
2.0    4
3.0    3
0.0    3
8.0    3
Name: Final Pool, dtype: int64
```

```
# get x and y values from result
type(result)
```

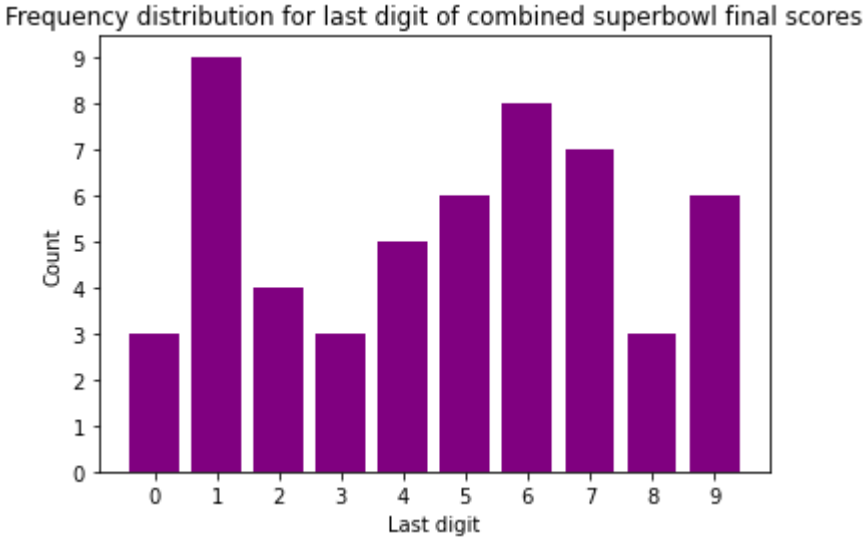
```
pandas.core.series.Series
```

```
print("column 1", result.index)
print("column 2", result. values)
```

```
column 1 Float64Index([1.0, 6.0, 7.0, 9.0, 5.0, 4.0, 2.0, 3.0, 0.0, 8.0], dtype='float64
```

```
column 2 [9 8 7 6 6 5 4 3 3 3]
```

```
# Make a beautiful bar chart with the data!!
plt.bar(result.index,result.values,width=.8,color='purple')
plt.title("Frequency distribution for last digit of combined superbowl final scores")
plt.xlabel("Last digit")
plt.ylabel("Count")
plt.xticks(np.arange(10))
plt.yticks(np.arange(10))
plt.show()
```



Conclusion

It seems that for the final combined scores, number 1 is the most commonly occuring last digit. Following 1 are 6 and then 7. The least common digits are a three-way tie between 0, 3, and 8, all have occurred 3 times in the past 54 superbowls. I am suprised with numbers 1 and 6 being the most common digits. I am not very familiar with the rules and scoring of football, but I know that the most common way of getting points is via a touchdown (6 pts) followed by a field goal (1 pt) for a total of 7 points.

Question 2: Querying a dataframe.

Create your own question about the superbowl data that you can answer using the Pandas dataframe and querying techniques. Some sample questions are below. You are welcome to pick one, but encouraged to create a question of interest to you.

Sample questions:

1. How many times has the team that was ahead at half time ended up winning the game?

2. How many times have the games been blowouts (the opposite of close)? You should decide on a definition of blowout and explain why you chose that definition.
3. How many times has one person won the whole pool? (I.e., when has the pool number for half time and final been the same?)
4. Or create your own question!

You should include commented code (with your own comments!) and a brief conclusion discussing what interested you about the question you chose and what you learned in your analysis of the data.

One Person Wins All

I thought it would be interesting to see how often one person wins both pools. To do this, I had to implement the code we wrote in class that finds the pool of the first half. Once I had that, it was easy to implement the query that compares the half pool to the final pool. I wanted to know the answer to this question because it would be great if the odds are good of winning both pools! Apparently it has only happened 12 times that someone has won both pools, so we shouldn't get our hopes up too high.

```
# How many times has one person won the whole pool?

# Adding column for the halftime
df['Combined Half']=df['Team1 1st']+df['Team1 2nd']+df['Team2 1st']+df['Team2 2nd']
df["half pool"]=df['Combined Half']%10
df.head()
```

```
# query database for when the half pool and the final pool are equal to each other
dfOneWinner=df[ df['Final Pool'] == df['half pool'] ]
dfOneWinner
```

Year	Team 1	Team1 1st	Team1 2nd	Team1 3rd	Team1 4th	Team1 Final	Team 2	Team2 1st	Team2 2nd	Team2 3rd	Te
2018	Philadelphia Eagles (NFC, 13-3)	9.0	13.0	7.0	12.0	41.0	New England Patriots (AFC, 13-3)	3.0	9.0	14.0	
2002	St. Louis Rams (NFC, 14-2)	3.0	0.0	0.0	14.0	17.0	New England Patriots (AFC, 11-5)	0.0	14.0	3.0	
2000	St. Louis Rams (NFC, 13-3)	3.0	6.0	7.0	7.0	23.0	Tennessee Titans (AFC, 13-3)	0.0	0.0	6.0	
1999	Denver Broncos (AFC, 14-2)	7.0	10.0	0.0	17.0	34.0	Atlanta Falcons (NFC, 14-2)	3.0	3.0	0.0	
1989	Cincinnati Bengals (AFC, 12-4)	0.0	3.0	10.0	3.0	16.0	San Francisco 49ers (NFC, 10-6)	3.0	0.0	3.0	
1987	Denver Broncos (AFC, 11-5)	10.0	0.0	0.0	10.0	20.0	New York Giants (NFC, 14-2)	7.0	2.0	17.0	
1986	Chicago Bears (NFC, 15-1)	13.0	10.0	21.0	2.0	46.0	New England Patriots (AFC, 11-5)	3.0	0.0	0.0	
1985	Miami Dolphins	10.0	6.0	0.0	0.0	16.0	San Francisco	7.0	21.0	10.0	

```
# Print the number of how many times one person has won the whole pool
dfOneWinner.shape[0]
```

```
12
```