

Project 3 Solutions

Kevin Choe

Collaborators: (Collaborators listed here. Include names, which part of the project you gave or sought help with, and how you helped or were helped.)

TA help: Summeth Guda

Online resources used: (List of links/resources (if any) here. Include web addresses, which part of the project the resource helped with, and how you were helped.)

Question 1

```
import pandas as pd
myDF = pd.read_csv("/class/datamine/data/craigslist/vehicles.csv")
myDF.head()
```

```
      id  ...      long
0  7119256118  ... -114.2690
1  7120880186  ... -123.8240
2  7115048251  ...  -81.9654
3  7119250502  ... -114.2710
4  7120433904  ...  -68.8963
```

[5 rows x 25 columns]

Question 2

```
myyears = myDF['year'].dropna().to_list()
```

```
# get a list containing each unique year
unique_years = list(set(myyears))
```

```
# for each year (key), initialize the value (value) to 0
my_dict = {}
for year in unique_years:
    my_dict[year] = 0
for year in myyears:
    my_dict[year] = my_dict[year] + 1
print(my_dict)
```

```
{1900.0: 58, 1911.0: 1, 1912.0: 5, 1913.0: 1, 1915.0: 1, 1916.0: 3, 1917.0: 1,
1919.0: 2, 1920.0: 1, 1922.0: 4, 1923.0: 37, 1924.0: 8, 1925.0: 9, 1926.0: 11,
1927.0: 34, 1928.0: 27, 1929.0: 69, 1930.0: 53, 1931.0: 60, 1932.0: 39, 1933.0:
17, 1934.0: 44, 1935.0: 18, 1936.0: 50, 1937.0: 72, 1938.0: 22, 1939.0: 60,
1940.0: 94, 1941.0: 60, 1942.0: 22, 1943.0: 1, 1944.0: 3, 1945.0: 1, 1946.0:
79, 1947.0: 63, 1948.0: 101, 1949.0: 113, 1950.0: 120, 1951.0: 113, 1952.0: 88,
```

```
1953.0: 93, 1954.0: 94, 1955.0: 216, 1956.0: 188, 1957.0: 218, 1958.0: 88,
1959.0: 100, 1960.0: 103, 1961.0: 87, 1962.0: 170, 1963.0: 214, 1964.0: 299,
1965.0: 376, 1966.0: 490, 1967.0: 427, 1968.0: 415, 1969.0: 464, 1970.0: 412,
1971.0: 383, 1972.0: 427, 1973.0: 392, 1974.0: 305, 1975.0: 219, 1976.0: 266,
1977.0: 313, 1978.0: 408, 1979.0: 468, 1980.0: 263, 1981.0: 219, 1982.0: 185,
1983.0: 243, 1984.0: 430, 1985.0: 477, 1986.0: 499, 1987.0: 635, 1988.0: 539,
1989.0: 630, 1990.0: 646, 1991.0: 649, 1992.0: 762, 1993.0: 825, 1994.0: 1113,
1995.0: 1433, 1996.0: 1529, 1997.0: 2062, 1998.0: 2351, 1999.0: 3742, 2000.0:
4778, 2001.0: 5742, 2002.0: 7153, 2003.0: 9235, 2004.0: 12037, 2005.0: 14209,
2006.0: 17191, 2007.0: 20457, 2008.0: 22643, 2009.0: 15899, 2010.0: 19956,
2011.0: 26532, 2012.0: 29108, 2013.0: 31434, 2014.0: 31703, 2015.0: 32918,
2016.0: 32096, 2017.0: 34592, 2018.0: 20147, 2019.0: 15531, 2020.0: 2820,
2021.0: 119}
```

Question 3

```
import matplotlib.pyplot as plt
```

```
my_dict.keys()
```

```
dict_keys([1900.0, 1911.0, 1912.0, 1913.0, 1915.0, 1916.0, 1917.0, 1919.0,
1920.0, 1922.0, 1923.0, 1924.0, 1925.0, 1926.0, 1927.0, 1928.0, 1929.0, 1930.0,
1931.0, 1932.0, 1933.0, 1934.0, 1935.0, 1936.0, 1937.0, 1938.0, 1939.0, 1940.0,
1941.0, 1942.0, 1943.0, 1944.0, 1945.0, 1946.0, 1947.0, 1948.0, 1949.0, 1950.0,
1951.0, 1952.0, 1953.0, 1954.0, 1955.0, 1956.0, 1957.0, 1958.0, 1959.0, 1960.0,
1961.0, 1962.0, 1963.0, 1964.0, 1965.0, 1966.0, 1967.0, 1968.0, 1969.0, 1970.0,
1971.0, 1972.0, 1973.0, 1974.0, 1975.0, 1976.0, 1977.0, 1978.0, 1979.0, 1980.0,
1981.0, 1982.0, 1983.0, 1984.0, 1985.0, 1986.0, 1987.0, 1988.0, 1989.0, 1990.0,
1991.0, 1992.0, 1993.0, 1994.0, 1995.0, 1996.0, 1997.0, 1998.0, 1999.0, 2000.0,
2001.0, 2002.0, 2003.0, 2004.0, 2005.0, 2006.0, 2007.0, 2008.0, 2009.0, 2010.0,
2011.0, 2012.0, 2013.0, 2014.0, 2015.0, 2016.0, 2017.0, 2018.0, 2019.0, 2020.0,
2021.0])
```

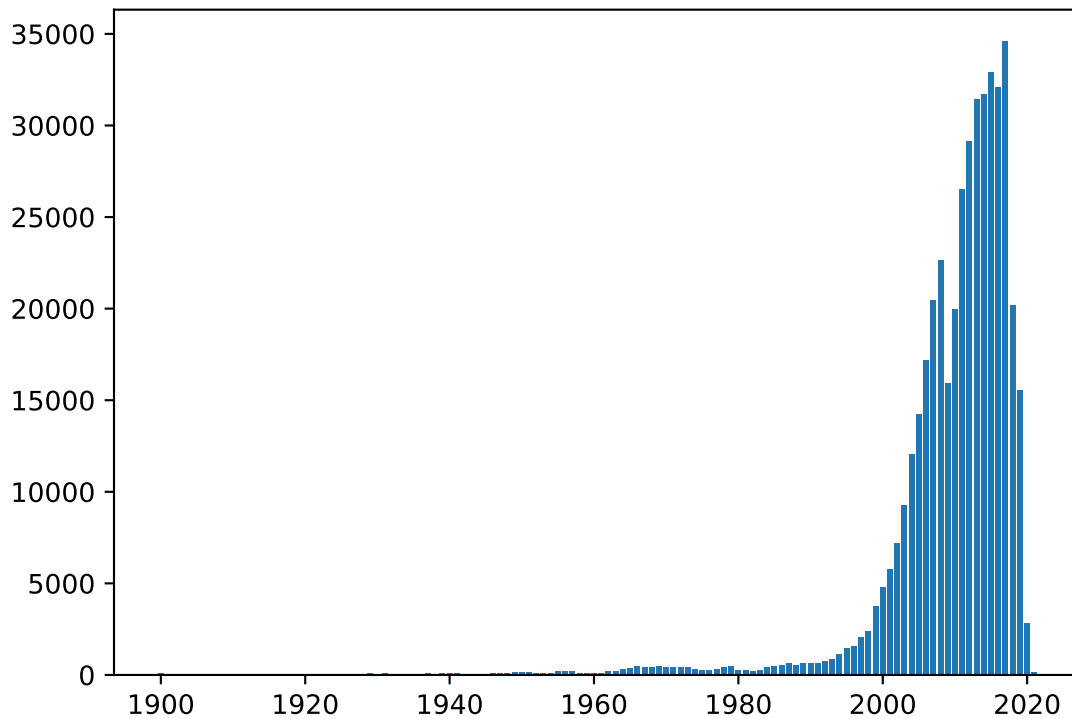
```
my_dict.values()
```

```
dict_values([58, 1, 5, 1, 1, 3, 1, 2, 1, 4, 37, 8, 9, 11, 34, 27, 69, 53, 60,
39, 17, 44, 18, 50, 72, 22, 60, 94, 60, 22, 1, 3, 1, 79, 63, 101, 113, 120,
113, 88, 93, 94, 216, 188, 218, 88, 100, 103, 87, 170, 214, 299, 376, 490, 427,
415, 464, 412, 383, 427, 392, 305, 219, 266, 313, 408, 468, 263, 219, 185, 243,
430, 477, 499, 635, 539, 630, 646, 649, 762, 825, 1113, 1433, 1529, 2062, 2351,
3742, 4778, 5742, 7153, 9235, 12037, 14209, 17191, 20457, 22643, 15899, 19956,
26532, 29108, 31434, 31703, 32918, 32096, 34592, 20147, 15531, 2820, 119])
```

```
plt.bar(my_dict.keys(), my_dict.values())
```

```
<BarContainer object of 109 artists>
```

```
plt.show()
```



```
plt.close()
# An example of a key would be 1912 has a value of 5.
print(my_dict[1912])
```

5

Question 4

```
listA = [1, 2, 3, 4, 5, 6, 12, 12]
listB = [2, 1, 7, 7, 7, 2, 8, 9, 10, 11, 12, 13]
```

```
# 1. values in list A but not list B
# values in list A but not list B
print(set(listA) - set(listB))
```

```
# 2. values in listB but not list A
```

```
{3, 4, 5, 6}
```

```
print(set(listB) - set(listA))
```

```
# 3. values in both lists
# values in both lists
```

```
{7, 8, 9, 10, 11, 13}
```

```
print(set.intersection(set(listA),set(listB)))
```

```
{1, 2, 12}
```

Question 5

```
states_list = list(myDF.loc[:, ["state", "lat", "long"]].dropna().to_records(index=False))

geoDict = {}
for my_triple in states_list:
    geoDict[my_triple[0]] = []

for i in states_list:
    geoDict[i[0]].append( (i[1], i[2]) )
```

Question 6

```
import matplotlib.pyplot as plt
```

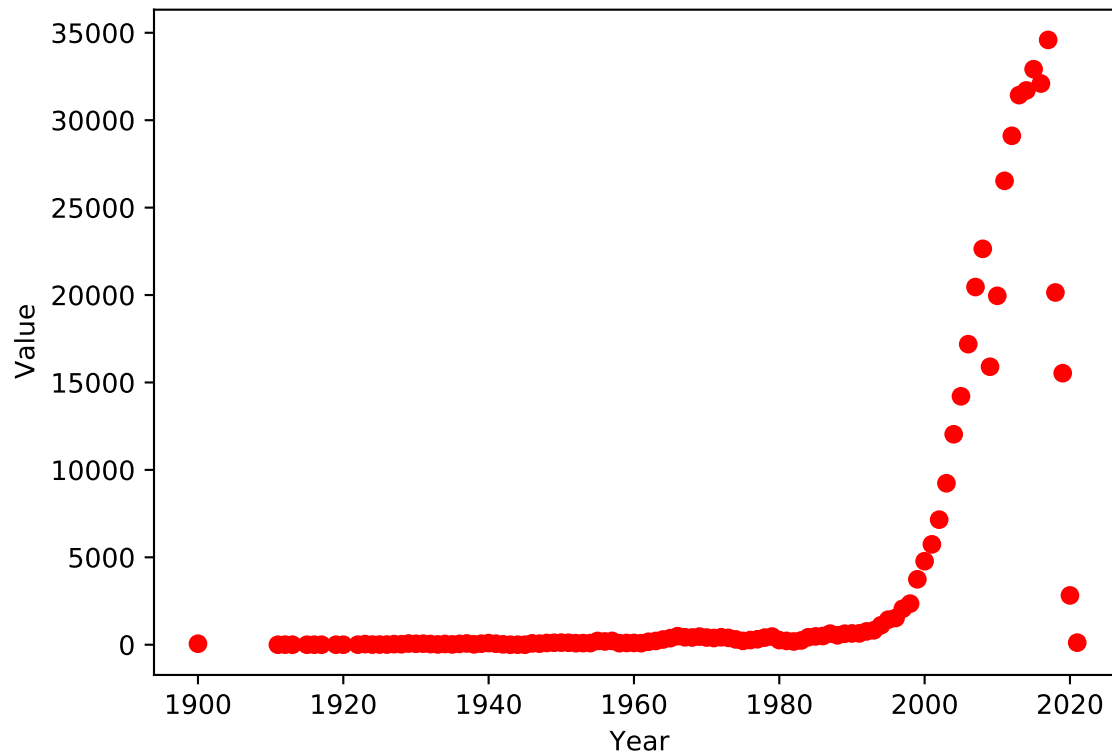
```
my_dict.keys()
```

```
dict_keys([1900.0, 1911.0, 1912.0, 1913.0, 1915.0, 1916.0, 1917.0, 1919.0,
1920.0, 1922.0, 1923.0, 1924.0, 1925.0, 1926.0, 1927.0, 1928.0, 1929.0, 1930.0,
1931.0, 1932.0, 1933.0, 1934.0, 1935.0, 1936.0, 1937.0, 1938.0, 1939.0, 1940.0,
1941.0, 1942.0, 1943.0, 1944.0, 1945.0, 1946.0, 1947.0, 1948.0, 1949.0, 1950.0,
1951.0, 1952.0, 1953.0, 1954.0, 1955.0, 1956.0, 1957.0, 1958.0, 1959.0, 1960.0,
1961.0, 1962.0, 1963.0, 1964.0, 1965.0, 1966.0, 1967.0, 1968.0, 1969.0, 1970.0,
1971.0, 1972.0, 1973.0, 1974.0, 1975.0, 1976.0, 1977.0, 1978.0, 1979.0, 1980.0,
1981.0, 1982.0, 1983.0, 1984.0, 1985.0, 1986.0, 1987.0, 1988.0, 1989.0, 1990.0,
1991.0, 1992.0, 1993.0, 1994.0, 1995.0, 1996.0, 1997.0, 1998.0, 1999.0, 2000.0,
2001.0, 2002.0, 2003.0, 2004.0, 2005.0, 2006.0, 2007.0, 2008.0, 2009.0, 2010.0,
2011.0, 2012.0, 2013.0, 2014.0, 2015.0, 2016.0, 2017.0, 2018.0, 2019.0, 2020.0,
2021.0])
```

```
my_dict.values()
```

```
dict_values([58, 1, 5, 1, 1, 3, 1, 2, 1, 4, 37, 8, 9, 11, 34, 27, 69, 53, 60,
39, 17, 44, 18, 50, 72, 22, 60, 94, 60, 22, 1, 3, 1, 79, 63, 101, 113, 120,
113, 88, 93, 94, 216, 188, 218, 88, 100, 103, 87, 170, 214, 299, 376, 490, 427,
415, 464, 412, 383, 427, 392, 305, 219, 266, 313, 408, 468, 263, 219, 185, 243,
430, 477, 499, 635, 539, 630, 646, 649, 762, 825, 1113, 1433, 1529, 2062, 2351,
3742, 4778, 5742, 7153, 9235, 12037, 14209, 17191, 20457, 22643, 15899, 19956,
26532, 29108, 31434, 31703, 32918, 32096, 34592, 20147, 15531, 2820, 119])
```

```
plt.scatter(my_dict.keys(), my_dict.values(), color='red')
plt.xlabel("Year")
plt.ylabel("Value");
plt.show()
```



```
plt.close()
```

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do.
Accountable together – We are Purdue.