Project 02 Demo Solution - Ewen Dai

March 7, 2019

```
In [3]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

1 Project 02 Demo Solution

This is a sample mini-solution for Project 02 of the DSS Decal during Spring 2019.

1.0.1 Importing the Dataset

The dataset I will be using is pulled from data.gov. (https://catalog.data.gov/dataset/school-improvement-2010-grants)

This is the School Improvement 2010 Grants as provided by the Department of Education.

Here is the summary of the data, should you be interested:

"Since President Obama took office, Congress has appropriated more than \$4 billion to help turn around the nation's lowest-performing schools. States were awarded nearly \$3.5 billion in School Improvement Grant funds in 2010 to turn around their persistently lowest achieving schools. School districts then applied to state for the funds this spring. When school districts applied, they were required to indicate that they would implement one of the following four models in their persistently lowest achieving schools: - Turnaround Model: Replace the principal, screen existing school staff, and rehire no more than half the teachers; adopt a new governance structure; and improve the school through curriculum reform, professional development, extending learning time, and other strategies. - Restart Model: Convert a school or close it and re-open it as a charter school or under an education management organization. - School Closure: Close the school and send the students to higher-achieving schools in the district. - Transformation Model: Replace the principal and improve the school through comprehensive curriculum reform, professional development, extending learning time, and other strategies."

I downloaded the csv with my data and put it in the same folder as the iPython Notebook I am working on so I can access it:

```
In [4]: ls
```

```
'Project 02 Demo Solution - Ewen Dai.ipynb'
 userssharedsdfschoolimprovement2010grants.csv
```

I will now load the file into a pandas data frame:

```
In [5]: grants2010 = pd.read_csv('userssharedsdfschoolimprovement2010grants.csv')
        grants2010.head()
```

```
Out[5]:                          School Name      City State  \
        0  HOGARTH KINGEEKUK MEMORIAL SCHOOL  SAVOONGA    AK
        1                    AKIACHAK SCHOOL  AKIACHAK    AK
        2                     GAMBELL SCHOOL   GAMBELL    AK
        3               BURCHELL HIGH SCHOOL   WASILLA    AK
        4                       AKIAK SCHOOL     AKIAK    AK

                                       District Name 2010/11/Award Amount  \
        0              BERING STRAIT SCHOOL DISTRICT           $471014.00
        1                     YUPIIT SCHOOL DISTRICT           $520579.00
        2              BERING STRAIT SCHOOL DISTRICT           $449592.00
        3  MATANUSKA-SUSITNA BOROUGH SCHOOL DISTRICT           $641184.00
        4                     YUPIIT SCHOOL DISTRICT           $399686.00

           Model Selected                                          Location
        0  Transformation  200 MAIN ST\nSAVOONGA, AK 99769\n(63.6687, -17...
        1  Transformation  AKIACHAK 51100\nAKIACHAK, AK 99551\n(60.8911, ...
        2  Transformation  169 MAIN ST\nGAMBELL, AK 99742\n(63.7413, -171...
        3  Transformation  1775 WEST PARKS HWY\nWASILLA, AK 99654\n(61.57...
        4  Transformation     AKIAK 5227\nAKIAK, AK 99552\n(60.8879, -161.2)
```

1.0.2 Looking at the Data

What are some questions that we might be able to answer using this data? What are some possible patterns that we might be curious about included within this data?

These are a few questions that you can consider to help you come up with some ideas of what to visualize in your graphs/charts/plots/etc.

For this demo solution, I will be graphing:

- the Total Awarded Amount ($) per state
- the Distributions of Models Selected
- the Number of schools per state vs. Total Amount Awarded

1.0.3 Basic Data Cleaning

This is not required for project 02, but is good practice and, in the long run, will make your life easier.

```
In [6]: # Change the column names to allow for easier reading and
        # extraction of data values
        grants2010.rename(columns = {"2010/11/Award Amount": "Award_Amount",
                                     "Model Selected": "Model"}, inplace = True)

        # Change the values of the Award_Amount to numericals
        type(grants2010['Award_Amount'][0])
        # Notice that this tells us that the values are strings!!
        # Always pay attention to the data type of your data values

        grants2010['Award_Amount'] = grants2010['Award_Amount'].str.replace(pat = r'$',
                                                                             repl = r'')
        grants2010['Award_Amount'] = pd.to_numeric(grants2010['Award_Amount'])
        # Don't worry if you don't understand the above two lines:
        # this is just convenient for data cleaning, and not in the scope of the course.

        grants2010.head()
```

```
Out[6]:                           School Name      City State  \
        0  HOGARTH KINGEEKUK MEMORIAL SCHOOL  SAVOONGA    AK
        1                    AKIACHAK SCHOOL  AKIACHAK    AK
        2                     GAMBELL SCHOOL   GAMBELL    AK
        3                BURCHELL HIGH SCHOOL   WASILLA    AK
        4                       AKIAK SCHOOL     AKIAK    AK

                                  District Name  Award_Amount           Model  \
        0              BERING STRAIT SCHOOL DISTRICT      471014.0  Transformation
        1                     YUPIIT SCHOOL DISTRICT      520579.0  Transformation
        2              BERING STRAIT SCHOOL DISTRICT      449592.0  Transformation
        3  MATANUSKA-SUSITNA BOROUGH SCHOOL DISTRICT      641184.0  Transformation
        4                     YUPIIT SCHOOL DISTRICT      399686.0  Transformation

                                                   Location
        0  200 MAIN ST\nSAVOONGA, AK 99769\n(63.6687, -17...
        1  AKIACHAK 51100\nAKIACHAK, AK 99551\n(60.8911, ...
        2  169 MAIN ST\nGAMBELL, AK 99742\n(63.7413, -171...
        3  1775 WEST PARKS HWY\nWASILLA, AK 99654\n(61.57...
        4     AKIAK 5227\nAKIAK, AK 99552\n(60.8879, -161.2)
```

Generally you would want to do a thorough cleaning of the data. For instance, continue by tidying up the values of the "Location" column. I will not be doing that here because in the three visuals I have planned to create, "Location" is not considered.

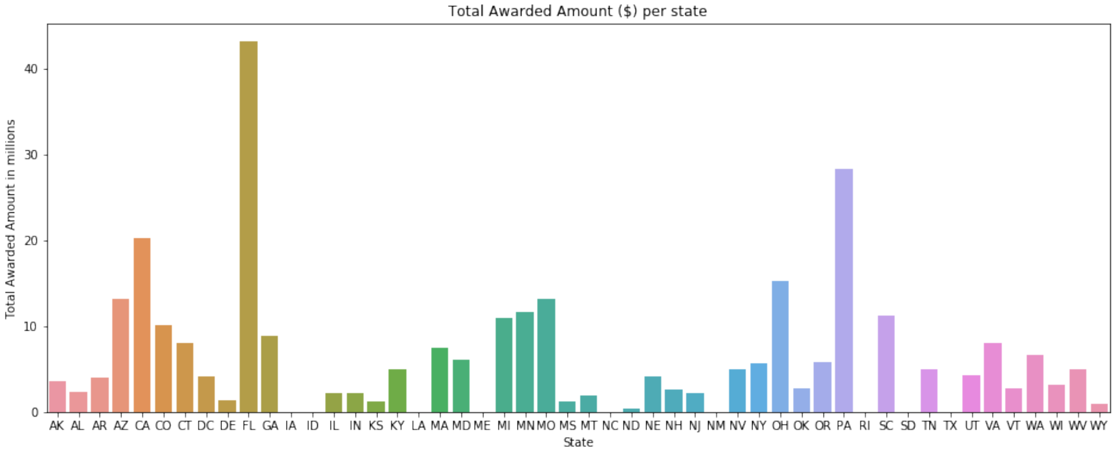
1.0.4 Total Awarded Amount ($) per state in millions

The data that we need are the Total Awarded Amounts per each individual state.

```
In [7]: awarded_amt = grants2010.loc[:,
                    ['State', 'Award_Amount']].groupby('State').agg(lambda x: sum(x)/1e6)
        awarded_amt.head()
```

```
Out[7]:        Award_Amount
        State
        AK         3.607416
        AL         2.300782
        AR         3.936109
        AZ        13.181345
        CA        20.286824
```

```
In [9]: plt.figure(figsize=(16, 6))
        sns.barplot(awarded_amt.index, awarded_amt['Award_Amount'])
        plt.title("Total Awarded Amount ($) per state")
        plt.xlabel("State")
        plt.ylabel("Total Awarded Amount in millions");
```

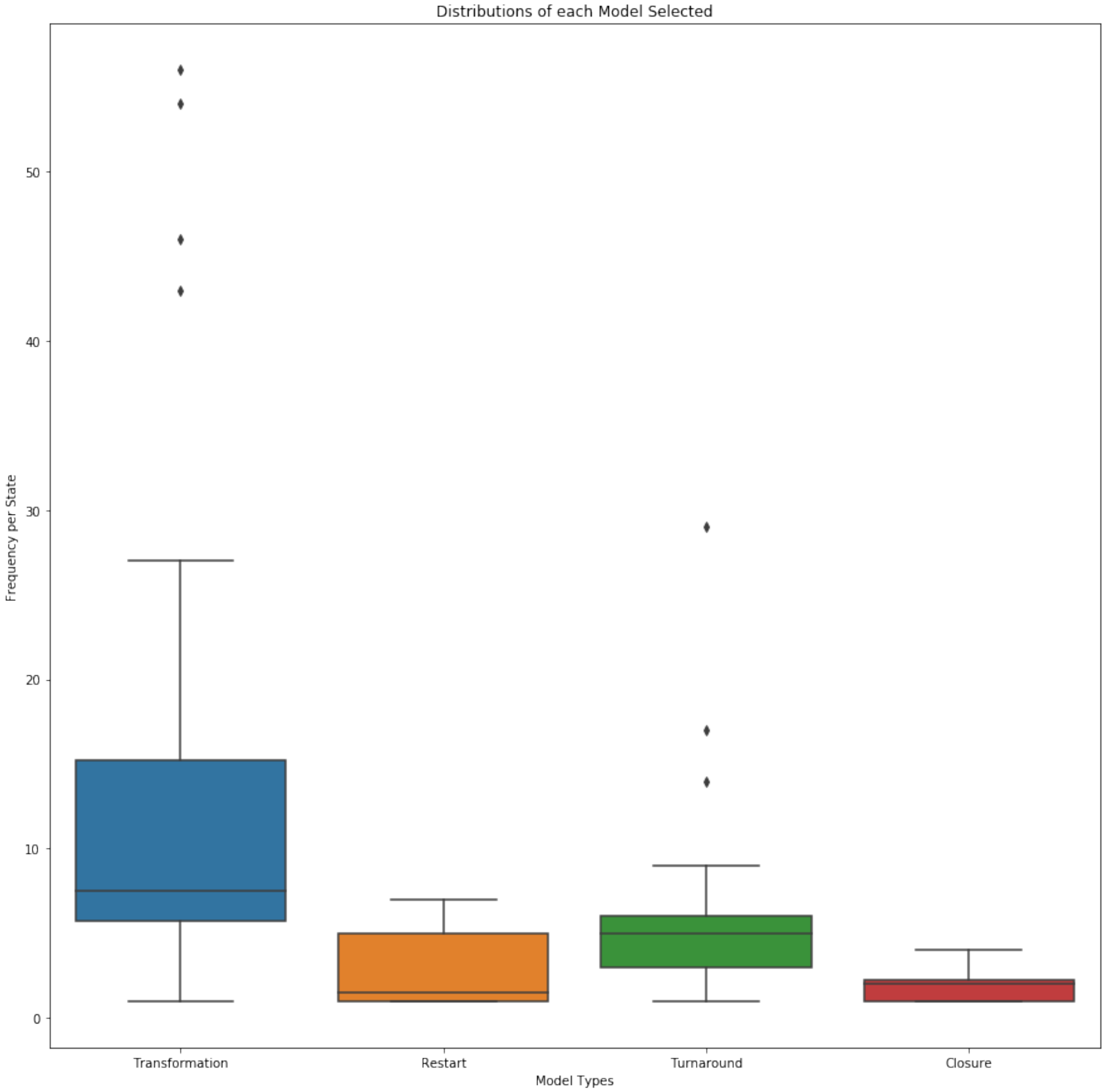


1.0.5 Distributions of Models Selected

```
In [10]: model = grants2010.loc[:, ['State',
                  'Model']].groupby(['Model', 'State']).agg(lambda x: len(x))
         model.head()

Out[10]: Model    State
         Closure  CA       2
                  CO       3
                  MO       1
                  PA       2
                  SC       1
         dtype: int64

In [11]: modeltypes = grants2010['Model'].unique()
         modeltypes

Out[11]: array(['Transformation', 'Restart', 'Turnaround', 'Closure', nan],
               dtype=object)

In [12]: congregatedData = {'types': [], 'values': []}
         for m in modeltypes[0:4]:
             for i in model[m]:
                 congregatedData['types'].append(m)
                 congregatedData['values'].append(i)

         modeldata = pd.DataFrame(congregatedData)
         modeldata.head()
```

```
Out[12]:             types  values
         0  Transformation       6
         1  Transformation      11
         2  Transformation       7
         3  Transformation      12
         4  Transformation      56
```

```
In [126]: plt.figure(figsize=(15, 15))
          sns.boxplot(x = 'types', y = 'values', data = modeldata)
          plt.title("Distributions of each Model Selected")
          plt.xlabel("Model Types")
          plt.ylabel("Frequency per State");
```

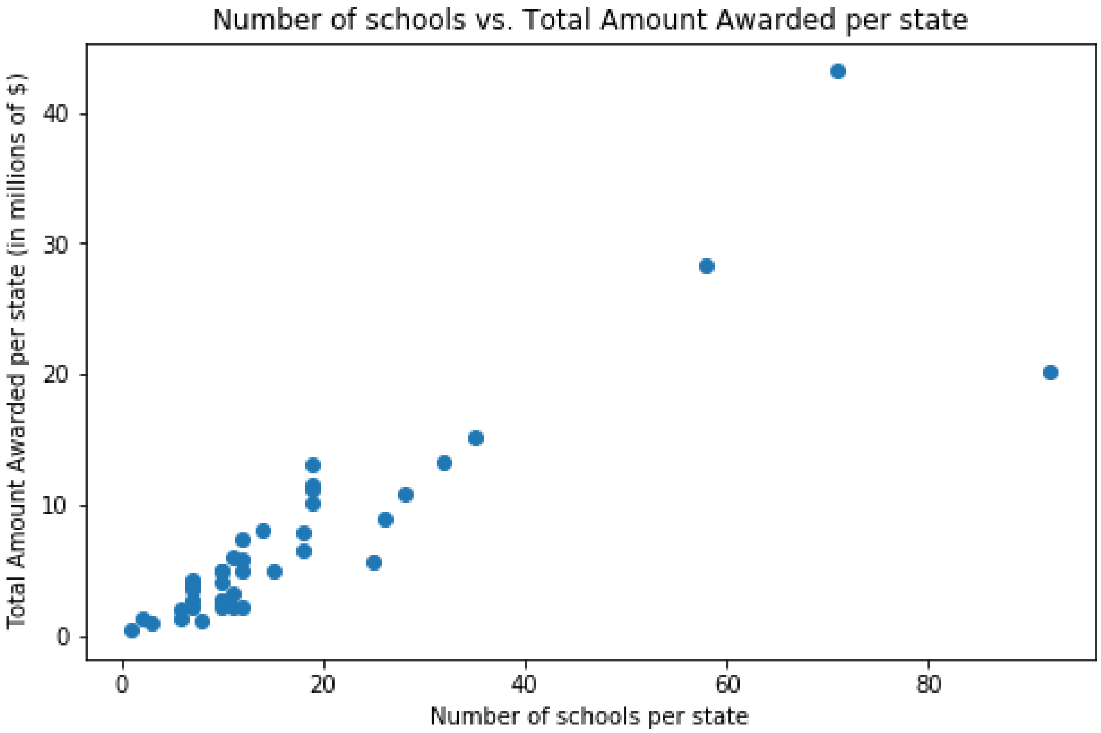


1.0.6 Number of schools vs. Total Amount Awarded per state

```
In [13]: school = grants2010.loc[:, ['State',
                 'School Name']].groupby('State').agg(lambda x: len(x))
         school.rename(columns = {"School Name": "Num_Schools"},
                       inplace = True)
         school['Award_Amount'] = awarded_amt['Award_Amount']
         school.head()
```

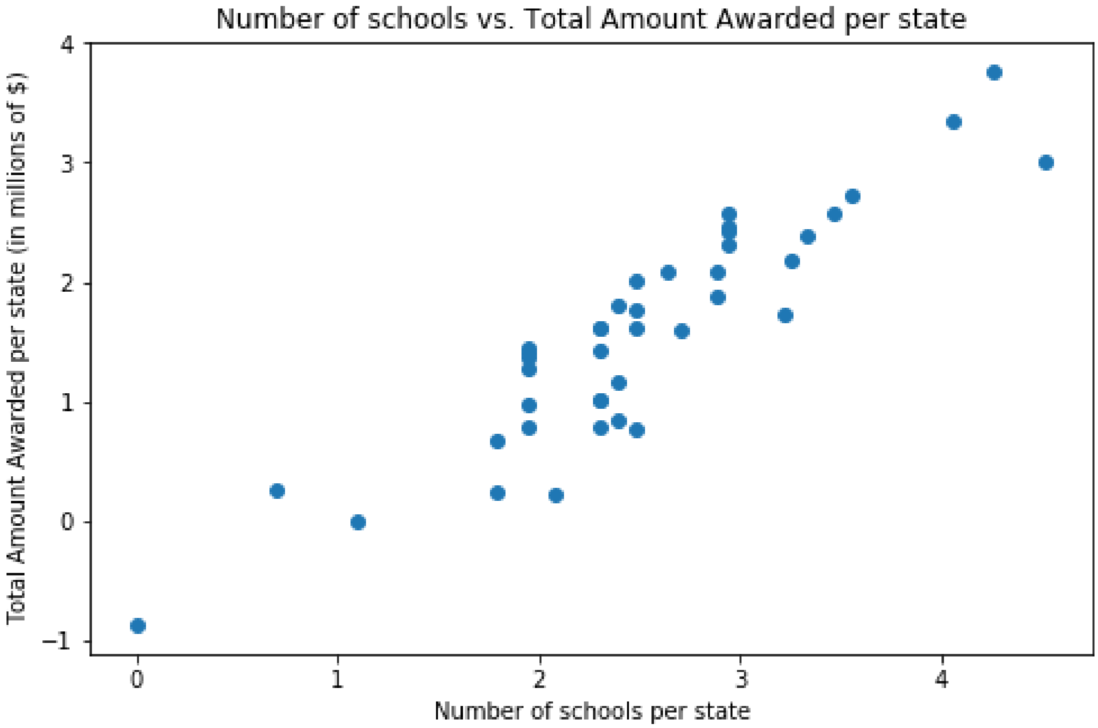
```
Out[13]:        Num_Schools  Award_Amount
         State
         AK               7      3.607416
         AL              11      2.300782
         AR               7      3.936109
         AZ              19     13.181345
         CA              92     20.286824
```

```
In [14]: plt.figure(figsize=(8, 5))
         plt.scatter(school['Num_Schools'], school['Award_Amount'])
         plt.title("Number of schools vs. Total Amount Awarded per state")
         plt.xlabel("Number of schools per state")
         plt.ylabel("Total Amount Awarded per state (in millions of $)");
```



There's some overplotting going on in this plot, so let's see if we can use `log` to make the points less crowded.

```
In [15]: plt.figure(figsize=(8, 5))
         plt.scatter(np.log(school['Num_Schools']), np.log(school['Award_Amount']))
         plt.title("Number of schools vs. Total Amount Awarded per state")
         plt.xlabel("Number of schools per state")
         plt.ylabel("Total Amount Awarded per state (in millions of $)");
```



That looks somewhat better. Doesn't it?

2 End Demo Solution.