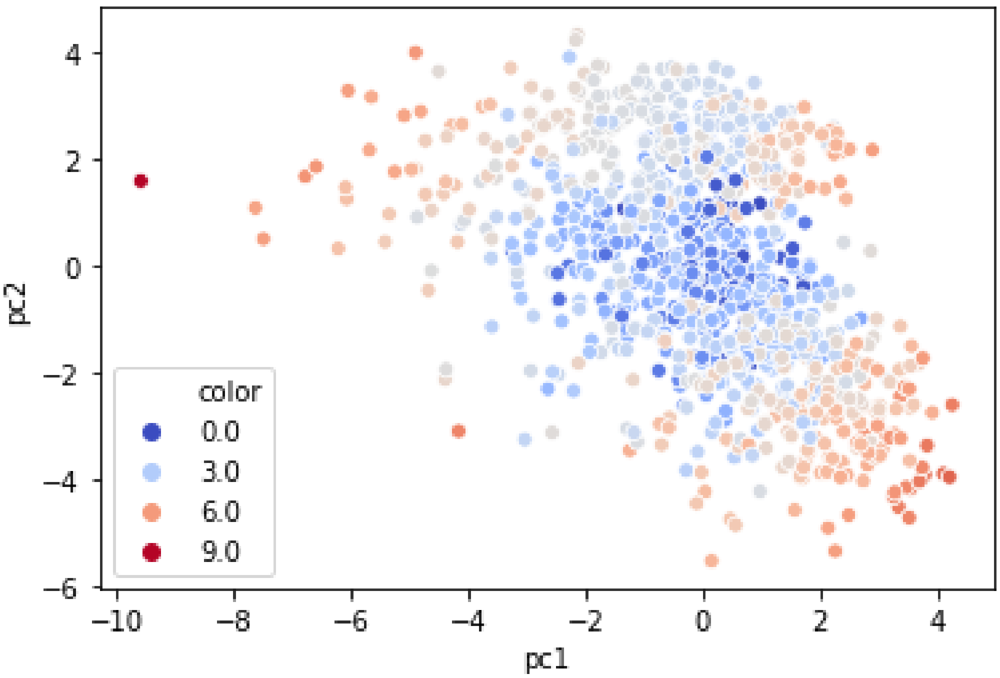
# Notebook

October 21, 2019

## 0.1 Question 2d

Create a 2D scatterplot of the first two principal components of `mid1_grades_centered_scaled`. Use `colorize_midterm_data` to add a `color` column to `mid1_1st_2_pcs`. Your code will be very similar to the code from problems 2a and 2b.

```
In [116]: u, s, vt = np.linalg.svd(mid1_grades_centered_scaled, full_matrices=False)
          d = {'pc1': mid1_grades_centered@vt.T[:,0], 'pc2': mid1_grades_centered@vt.T[:,1]}
          mid1_1st_2_pcs = pd.DataFrame(data=d)
          sns.scatterplot(data = colorize_midterm_data(mid1_1st_2_pcs), x = "pc1", y = "pc2", hue = "col
```

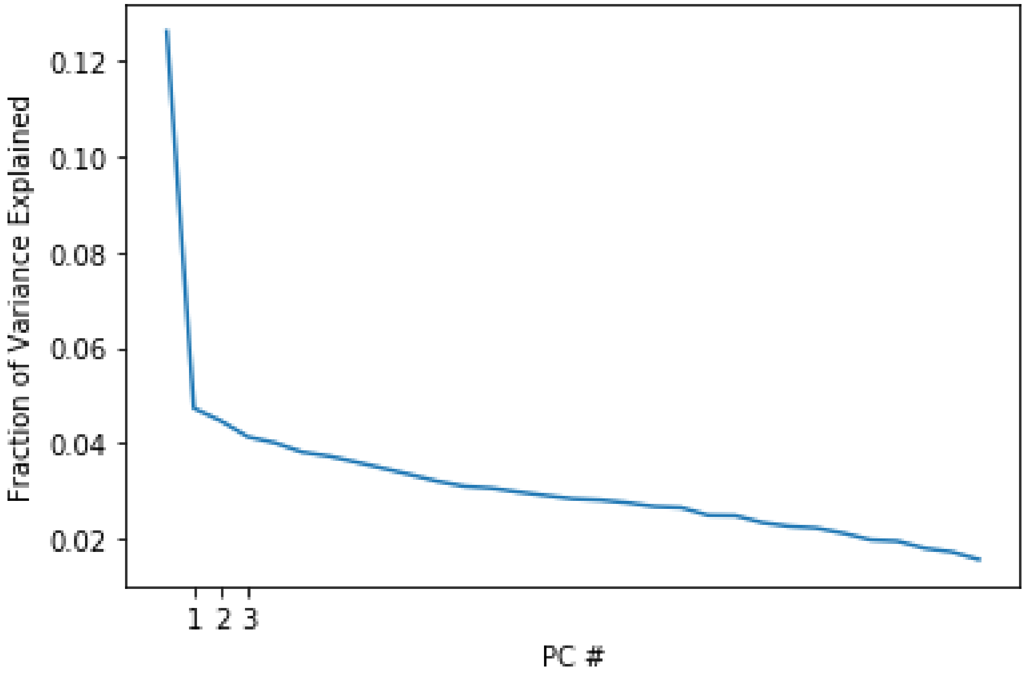


## 0.2 Question 2e

If you compute the fraction of the variance captured by this 2D scatter plot, you'll see it's only 17%, roughly 12% by the 1st PC, and roughly 5% by the 2nd PC. **In the cell below, create a scree plot showing the fraction of the variance explained by PC #i.**

Informally, we can say that our midterm scores matrix has a high rank. More formally, we can say that a rank 2 approximation only captures a small fraction of the variance, and thus the data are not particularly amenable to 2D PCA scatterplotting.

```
In [123]: plt.plot(s**2 / sum(s**2));
          plt.xticks([1, 2, 3],[1,2,3]);
          plt.xlabel('PC #');
          plt.ylabel('Fraction of Variance Explained');
```



Unfortunately, we have two problems:

1. There is a lot of overplotting, with only 27 distinct dots. This means that at least some states voted exactly alike in these elections.
2. We don't know which state is which, because the points are unlabeled.

Let's start by addressing problem 1.

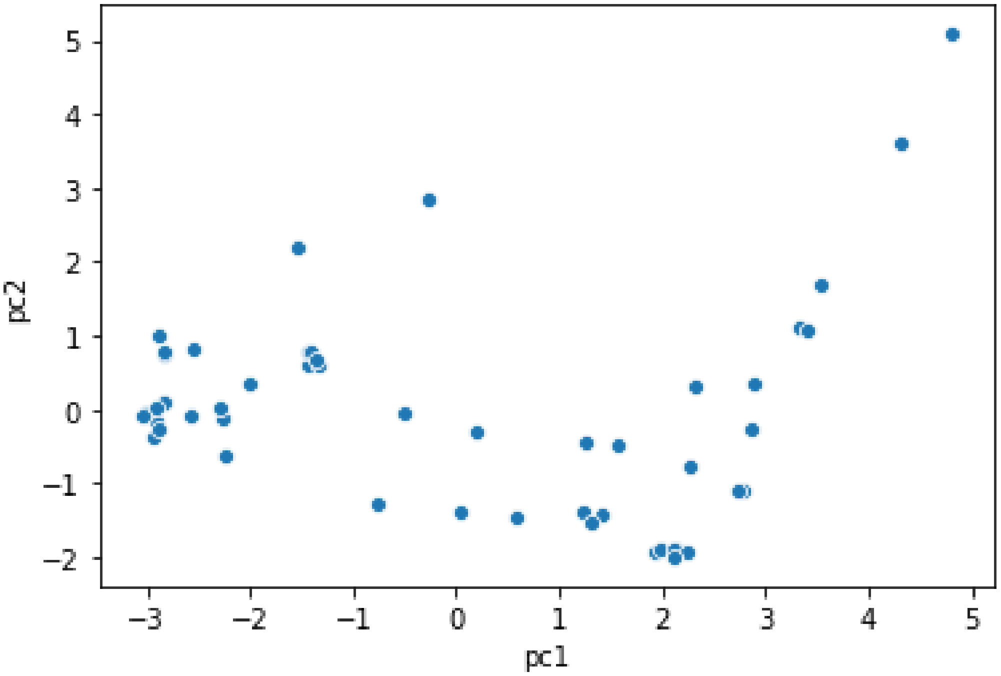
**In the cell below, create a new dataframe `first_2_pcs_jittered` with a small amount of random noise added to each principal component. In this same cell, create a scatterplot.**

The amount of noise you add should not significantly affect the appearance of the plot, it should simply serve to separate overlapping observations.

*Hint:* See the pairplot from the intro to question 2 for an example of how to introduce noise.

```
In [197]: first_2_pcs_jittered = first_2_pcs + np.random.normal(0, 0.1, size = (len(first_2_pcs), 2));
          sns.scatterplot(data = first_2_pcs_jittered, x = "pc1", y = "pc2");
```

```
Out[197]: State
          Alabama          -2.874640
          Alaska           -2.824363
          … Omitting 46 lines …
          Wisconsin         2.306688
          Wyoming          -3.051670
          Name: pc1, dtype: float64
```



## 0.3 Question 3g

To label our points, the best option is to use the `plotly` library that we used earlier in this homework. Plotly is an incredibly powerful plotting library that will automatically add axis labels, and will also provide controls so that you can zoom and pan around to look at the data.

One important skill as a user of modern tools is using existing documentation and examples to get the plot you want.

Using the example given on this page as a guide:, create a scatter plot of your **jittered data** from 3f with the following key properties:

1. Your plot should be created using `px.scatter`, and should use the `fig.update_traces` method to set the `textposition`.
2. Each point should be labeled by the name of the state, and the label should be above the point.

*Hint*: You can get a list of the state names with `list(df_1972_to_2016.index)`.

*Hint*: `gapminder` in the example linked is just the name of their dataframe. Your code shouldn't have anything to do with `gapminder` since we're plotting presidential election data, not life expectancies and gdp.

```
In [202]: import plotly.express as px
          new_data = {'State': df_1972_to_2016.index, 'pc1':first_2_pcs_jittered['pc1'],
                  'pc2':first_2_pcs_jittered['pc2']}
          new_data_frame = pd.DataFrame(new_data)
          fig = px.scatter(new_data_frame,  x = "pc1", y = "pc2", text="State", log_x=True, size_max=10
          fig.update_traces(textposition='top center')
          fig.show()
```

Give an example of a cluster of states that vote a similar way. Does the composition of this cluster surprise you? If you're not familiar with U.S. politics, it's fine to just say 'No, I'm not surprised because I don't know anything about U.S. politics.'.

A cluster of states that vote a similar way are states that are along the coast. For example, states like New York, New Jersey, Pennsylvania, Conneticut, Rhode Island, Maine, California, Oregon and Washington generally always vote democrat. The composition of the cluster does not surprise me because generally the states that vote similar are all clustered with each other.

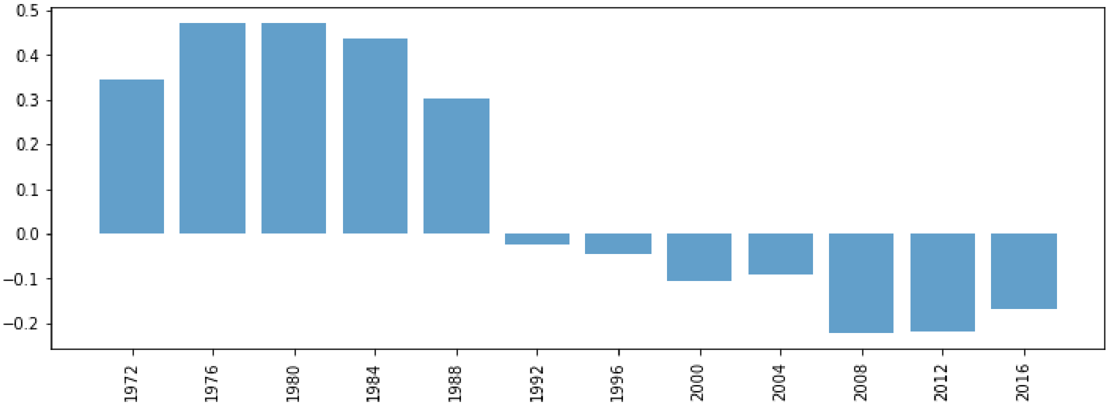
In the cell below, write down anything interesting that you observe by looking at this plot. You will get credit for this as long as you write something reasonable that you can take away from the plot.

Something interesting I observe from the plot is that Washington D.C and Minnesota are outliers are far off from the rest of the points. Looking at the dataframe from questino 3b) it seems this is the case because Washington D.C. and Minnesota have voted Democrat consistently from 1972 to 2016, where Minnesota only voted republican once in 1972.

In the cell below, plot the the 2nd row of $V^T$.

*Hint:* You are just copying and pasting code from the cell above and then changing one number.

```
In [204]: with plt.rc_context({"figure.figsize": (12, 4)}):
              plot_pc(list(df_1972_to_2016.columns), vt, 1);
```



## 0.4 Question 3i

Using your plots from question 3h as well as the original table, give a description of what it means to have a relatively large positive value for `pc1` (right side of the 2D scatter plot), and what it means to have a relatively large positive value for `pc2` (top side of the 2D scatter plot).

In other words, what is generally true about a state with relatively large positive value for `pc1`? For a large positive value for `pc2`?

Note: `pc2` is pretty hard to interpret, and the staff doesn't really have a concensus on what it means either. We'll be nice when grading.

Note: Principal components beyond the first are often hard to interpret (but not always, see question 1 earlier in this homework)

*Write your answer here, replacing this text.*

## 0.5 Question 3j

To get a better sense of whether our 2D scatterplot captures the whole story, create a scree plot for this data. On the y-axis plot the fraction of the total variance captured by the ith principal component. You should see that the first two principal components capture much more of the variance than we were able to capture when using the DS100 midterm 1 data. It is partially for this reason that the 2D scatter plot was so much more useful for this dataset.

*Hint:* Your code will be very similar to the scree plot from problem 1d. Be sure to label your axes appropriately!

```
In [ ]:
```