Data 100/200 Homework 11 Written

Jackie Hu

TOTAL POINTS

13 / 14

QUESTION 1

1 Question 2d 3 / 3

✓ **+ 3 pts Correct**

+ 2 pts Legend or axis titles missing.

+ 1 pts Data scatter incorrect, but otherwise correct code.

+ 0 pts Incorrect/blank

QUESTION 2

2 Question 2e 2 / 2

✓ **+ 2 pts Correct**

+ 1.5 pts Correct Plot

+ 0.5 pts Correct Labels and Title

+ 0 pts Blank/Incorrect

QUESTION 3

3 Question 3f 2 / 2

✓ **+ 2 pts Correct scatterplot with jittered points**

+ 1 pts Added noise, but no plot

+ 0 pts Incorrect/blank

QUESTION 4

Question 3g 2 pts

4.1 Part ii 1 / 1

✓ **+ 1 pts Any reasonable answer**

+ 0 pts Incorrect/Blank

4.2 Part iii 1 / 1

✓ **+ 1 pts Any reasonable observation**

+ 0 pts Incorrect/Blank

QUESTION 5

5 Question 3h 1 / 1

✓ **+ 1 pts Correct plot**

+ 0 pts Incorrect/blank

QUESTION 6

6 Question 3i 1 / 2

+ 2 pts Correct interpretation of pc1 and pc2

✓ **+ 1 pts Missing/Incorrect interpretation of either pc1 or pc2**

+ 0 pts Incorrect/Blank

QUESTION 7

7 Question 3j 2 / 2

✓ **+ 2 pts Correct**

+ 1.5 pts Correct Plot

+ 0.5 pts Correct Labels and Title

+ 0 pts Blank/Incorrect

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 [44]:
```
cntr_scaled = mid1_grades_centered_scaled - np.mean(mid1_grades_centered_scaled, axis = 0)
U_s,S_s,Vt_s = np.linalg.svd(cntr_scaled, full_matrices = False)
```

In [45]:
```
pcs_s = U_s @ np.diag(S_s)
pcs_s
```

Out[45]:
```
array([[-3.83072492, -0.7358655 ,  2.12611481, …, -0.78123605,
        -0.51686413,  0.85634594],
       [-1.8738641 ,  0.6962799 ,  0.47735479, …,  0.44245962,
        -0.79928174,  0.75248139],
       [ 1.74561932, -1.12581215,  0.14211213, …, -0.5789232 ,
         0.14050766,  0.05118241],
       …,
       [-0.65553033,  0.50128945,  1.29458714, …,  0.90518996,
        -0.61937273, -0.71177987],
       [ 1.00669349,  0.24188931,  0.39093521, …,  0.54326367,
         0.40351591, -0.10106796],
       [-4.17910477,  1.32156116,  0.89470089, …, -0.81809949,
         0.59998957,  0.38006272]])
```

In [46]:
```
pcs_df_s = pd.DataFrame(data = pcs_s)
mid1_2d_1st_2_pcs = pcs_df_s.iloc[:, :2]
mid1_2d_1st_2_pcs.rename(columns = {0:'pc1', 1:'pc2'}, inplace = True)
```

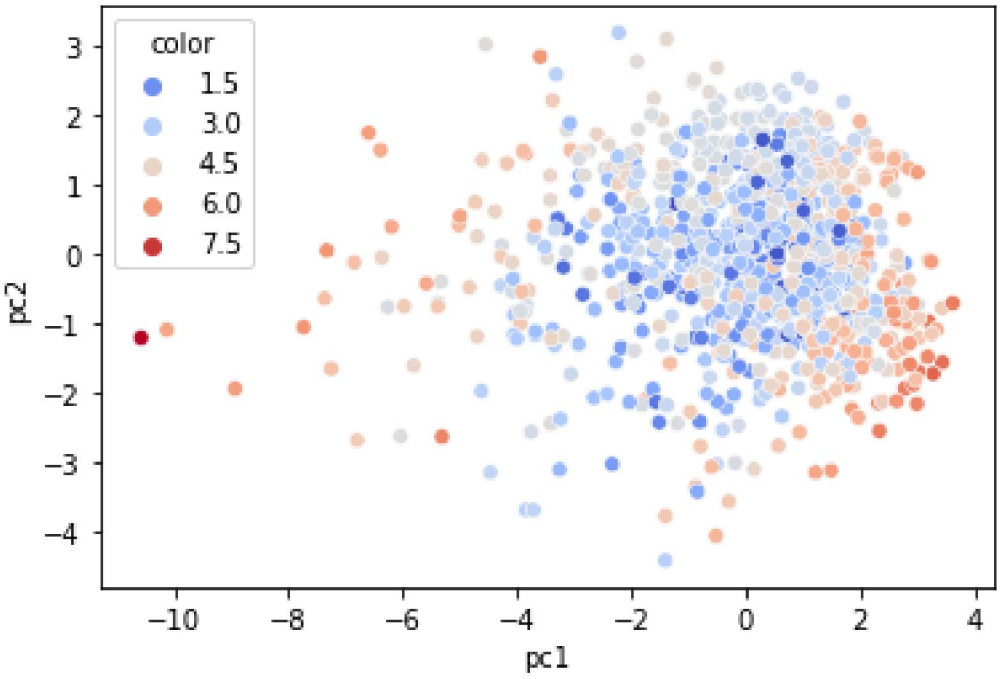
```
/opt/conda/lib/python3.8/site-packages/pandas/core/frame.py:4438: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
```

In [47]:
```
sns.scatterplot(data = colorize_midterm_data(mid1_2d_1st_2_pcs), x = "pc1", y = "pc2", hue = "
```

Out[47]:
```
<AxesSubplot:xlabel='pc1', ylabel='pc2'>
```

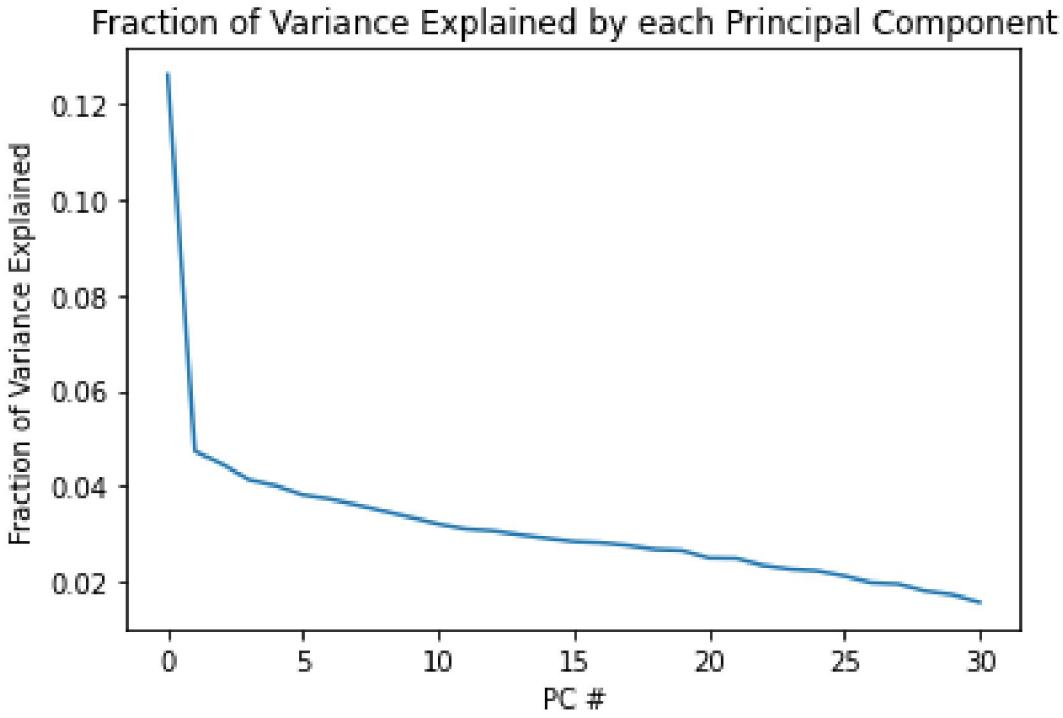

color
1.5
3.0
4.5
6.0
7.5
pc2
pc1
3
2
1
0
−1
−2
−3
−4
−10
−8
−6
−4
−2
0
2
4

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 each principle componant using the data from 2d.**

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 [108]: plt.plot( S_s**2 / sum(S_s**2));
          #plt.xticks([1, 2, 3], [1, 2, 3]);
          plt.xlabel('PC #');
          plt.ylabel('Fraction of Variance Explained');
          plt.title('Fraction of Variance Explained by each Principal Component');
```



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.

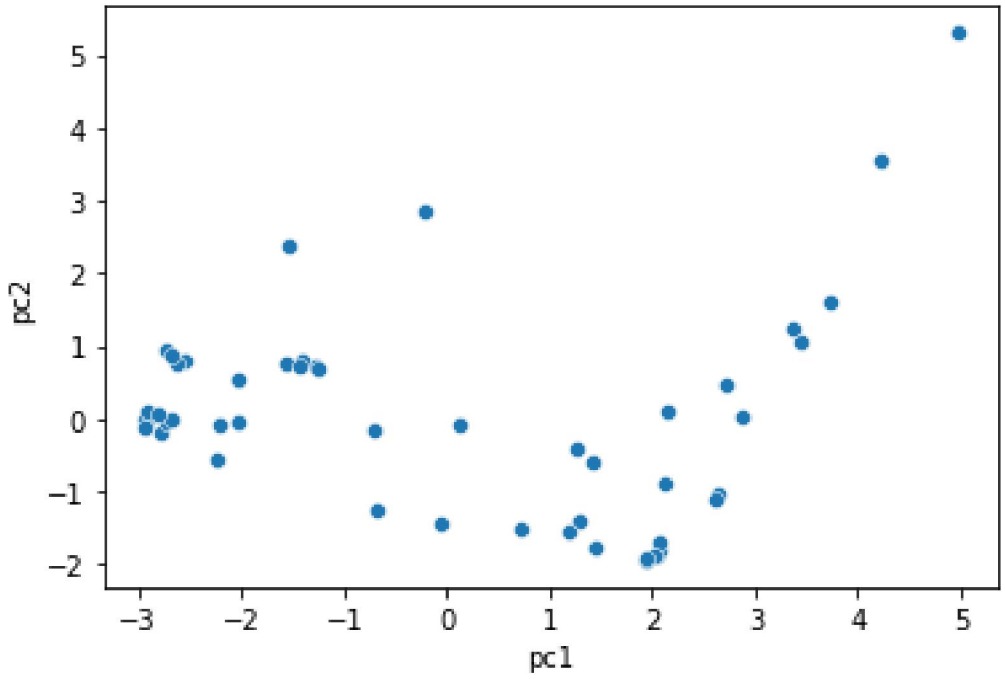
The amount of noise you add should not significantly affect the appearance of the plot, it should simply serve to separate overlapping observations. Don't get caught up on the exact details of your noise generation, it's fine as long as your plot looks roughly the same as the original scatterplot.

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

```
In [74]: first_2_pcs.head()
```

```
Out[74]:         pc1       pc2
         0 -2.733898  0.878935
         1 -2.854135 -0.068869
         2 -2.187073 -0.161736
         3 -1.399352  0.733236
         4  2.117444 -1.848357
```

```
In [75]: first_2_pcs_jittered = first_2_pcs.loc[:, 'pc1':'pc2'] + np.random.normal(0, 0.1, size = (len(
         sns.scatterplot(data = first_2_pcs_jittered, x = "pc1", y = "pc2");
```


pc2
pc1
5
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5

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.'.

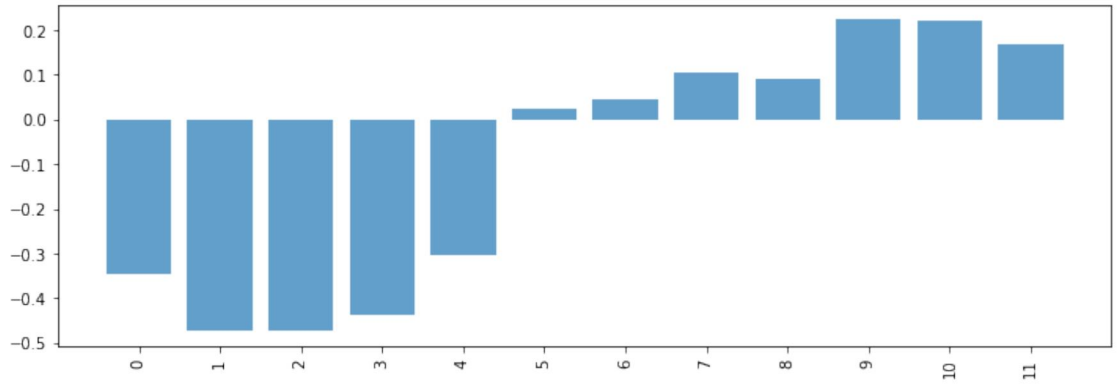
No, I'm not surprised because I don't know anything about U.S. politics.

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.

- The variables contributing similar information are grouped together, such as california, illinois, kennecticut, etc.
- When variables are negatively ("inversely") correlated, they are positioned on opposite sides of the plot origin, in diagonally quadrants. For instance, D.C. and Ohio,

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

```
In [80]: plt.figure(figsize=(12, 4))
         plot_pc(list(df_1972_to_2016.columns), vt_q3, 1);
```



0.3 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 consensus 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).

- A large positive value for pc1 means a stronger impact that variable has on the model, corresponding to a higher contribution for voting D.
- A large positive value for pc2 means that within the variance explained by the first conponent, a stronger impact that variable has on the second pc, which might be how different each year's contribution to the voting results for the same state.

0.4 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 Data 100 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 [109]: plt.plot(S_q3**2 / sum(S_q3**2));
          plt.xlabel('PC #');
          plt.ylabel('Fraction of Variance Explained');
          plt.title('Fraction of Variance Explained by each Principal Component');
```