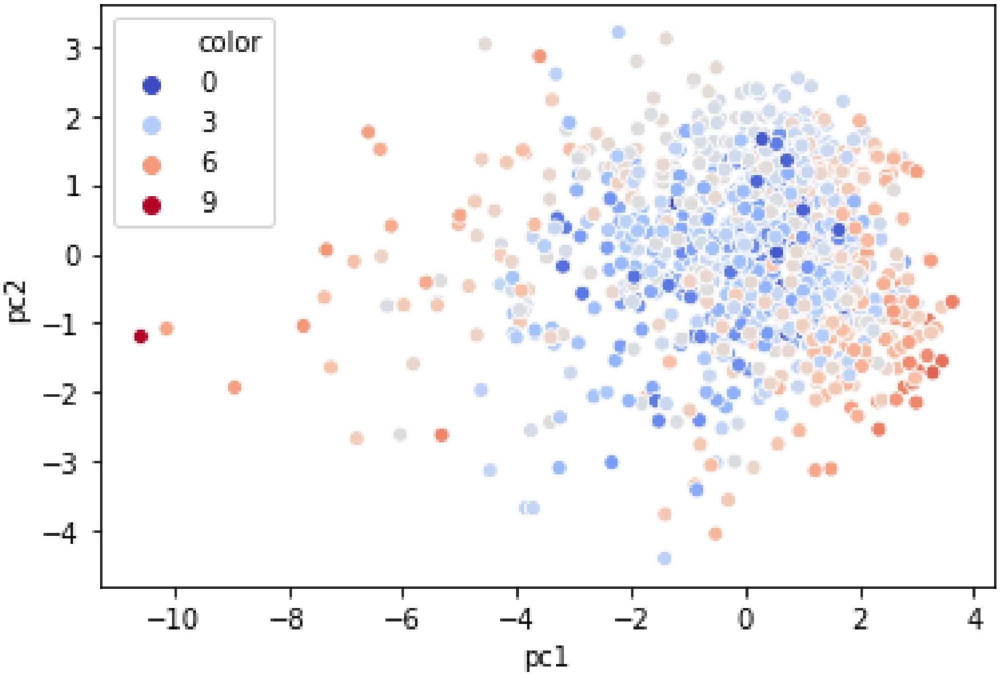
## 0.1 Question 2d

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_2d_1st_2_pcs`. Your code will be very similar to the code from problems 2a and 2b.

```
In [111]: u_2d, s_2d, vt_2d = np.linalg.svd(mid1_grades_centered_scaled, full_matrices=False)
          mid1_2d_1st_2_pcs = colorize_midterm_data(
              pd.DataFrame(
                  (u_2d @ np.diag(s_2d))[:, :2],
                  columns=["pc1", "pc2"]
              )
          )
          sns.scatterplot(data=mid1_2d_1st_2_pcs, x = "pc1", y = "pc2", hue = "color", palette = "coolwa
```



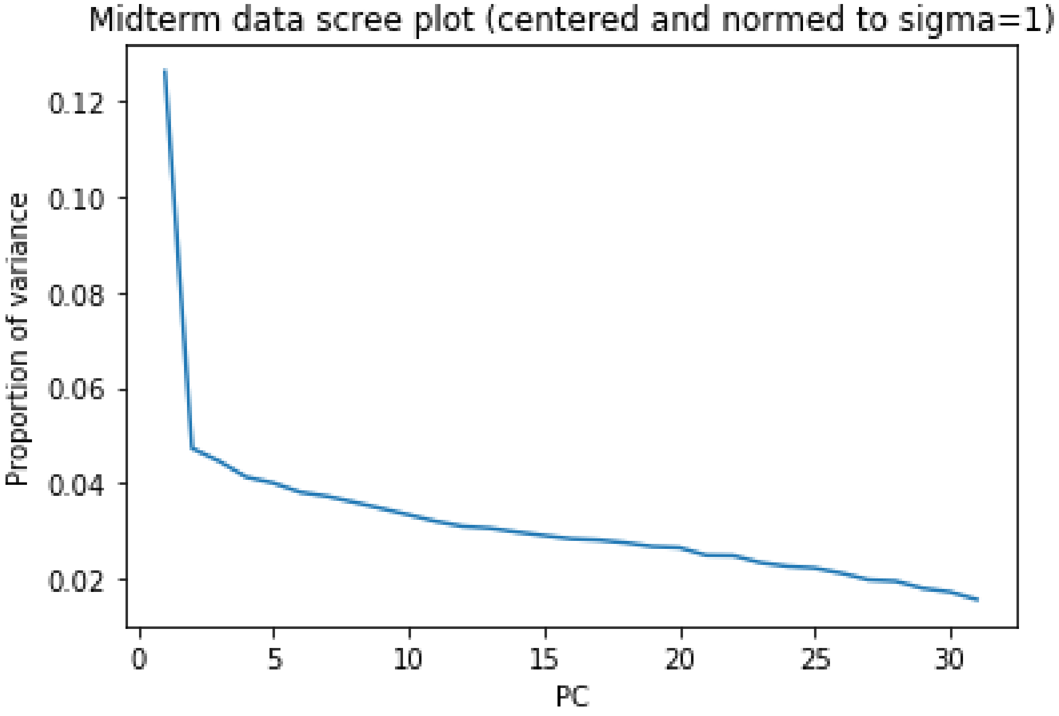
## 0.2 Question 2e

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 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 [115]: pc_i = np.arange(1, len(s_2d)+1)
          (sns.lineplot(x=pc_i, y=(s_2d**2) / sum(s_2d**2))
              .set(
                  title="Midterm data scree plot (centered and normed to sigma=1)",
                  xlabel="PC",
                  ylabel="Proportion of variance"
              )
          )
```

```
Out[115]: [Text(0, 0.5, 'Proportion of variance'),
           Text(0.5, 0, 'PC'),
           Text(0.5, 1.0, 'Midterm data scree plot (centered and normed to sigma=1)')]
```



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 [171]: first_2_pcs_jittered = first_2_pcs + np.random.normal(scale=0.5, size=first_2_pcs.shape)
          first_2_pcs_jittered = first_2_pcs_jittered.set_index(df_1972_to_2016.index)
          first_2_pcs_jittered
```

```
Out[171]:                      pc1       pc2
          State
          Alabama        -2.914103  0.431691
          Alaska         -4.012446  0.038387
          Arizona        -1.436266 -0.973962
          Arkansas       -1.104019  1.022620
          California      2.984962 -1.950580
          Colorado        0.127899 -0.647177
          Connecticut     1.861920 -1.798324
          Delaware        2.006768 -1.012420
          D.C.            5.200637  5.030347
          Florida        -0.172947  0.546886
          Georgia        -1.210591  2.087688
          Hawaii          2.929743  0.891096
          Idaho          -2.667713 -0.495738
          Illinois        1.754100 -2.572146
          Indiana        -2.356944 -1.128745
          Iowa            1.847095 -1.051720
          Kansas         -2.148184 -0.198641
          Kentucky       -1.107703 -0.112141
          Louisiana      -2.696834  0.064872
          Maine           1.731026 -1.716364
          Maryland        2.745532  0.467655
          Massachusetts   3.133140  1.174984
          Michigan        0.649083 -1.085500
          Minnesota       4.264931  2.843802
          Mississippi    -3.286721  1.250707
```

```
Missouri       -0.780596  0.689128
Montana        -1.646575 -0.074591
Nebraska       -2.876347 -0.364058
Nevada          1.256385 -1.734734
New Hampshire   1.585285 -1.340783
New Jersey      2.597766 -2.261354
New Mexico      2.121614 -1.303083
New York        2.177376  0.008761
North Carolina -1.952791  0.154888
North Dakota   -3.107192  0.174888
Ohio            0.290739  0.340372
Oklahoma       -2.656051 -0.442645
Oregon          3.225256 -1.293318
Pennsylvania    2.164482 -0.581275
Rhode Island    4.019371  0.996877
South Carolina -2.358751  1.647205
South Dakota   -3.087388  0.089796
Tennessee      -2.189708  0.807879
Texas          -2.694596  0.564710
Utah           -2.467891 -1.288114
Vermont         1.988537 -2.265601
Virginia       -1.353483 -1.320274
Washington      3.126122 -1.333610
West Virginia  -0.401660  2.673273
Wisconsin       3.114409  1.225630
Wyoming        -2.898676  0.008098
```

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.'.

One such cluster of states is North/South Dakota, Wyoming, Alabama, and Louisiana. This is unsurprising, as these states lean conservative and are not densely populated.

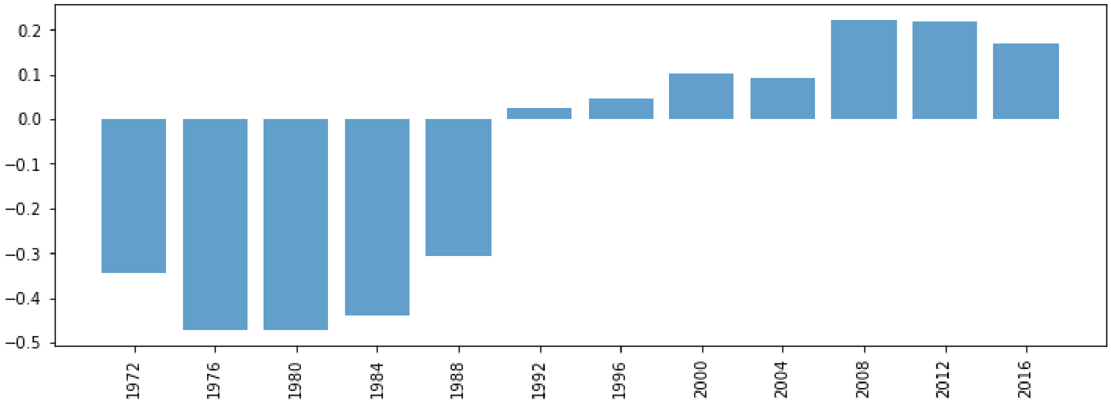
In the cell below, write down anything interesting that you observe by looking at this plot. You will get credit for this as long as you write something reasonable that you can take away from the plot.

It seems that generally speaking, `pc1` is an indication for the political leaning of certain states. Swing states like Michigan, Ohio, or Florida have `pc1` values near zero, while more right or left-leaning states such as Texas, Maryland, or California have higher magnitudes of `pc1`. In addition, positive `pc1` values seems to indicate a preference for Democrats while negative values indicate Republican preference.

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 [175]: with plt.rc_context({"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).

SOLUTION: Large positive values of `pc1` indicate that a state has a strong Democratic lean and tends to vote for liberal candidates. Large positive values of `pc2` may indicate something about population: D.C. which has the highest `pc2`, has a rather low population due to its small size; West Virginia, which also has a high `pc2`, is sparsely populated. Conversely, states with a low `pc2` include New Jersey, California, and Illinois, all of which are comapratively densely populated states with large cities.

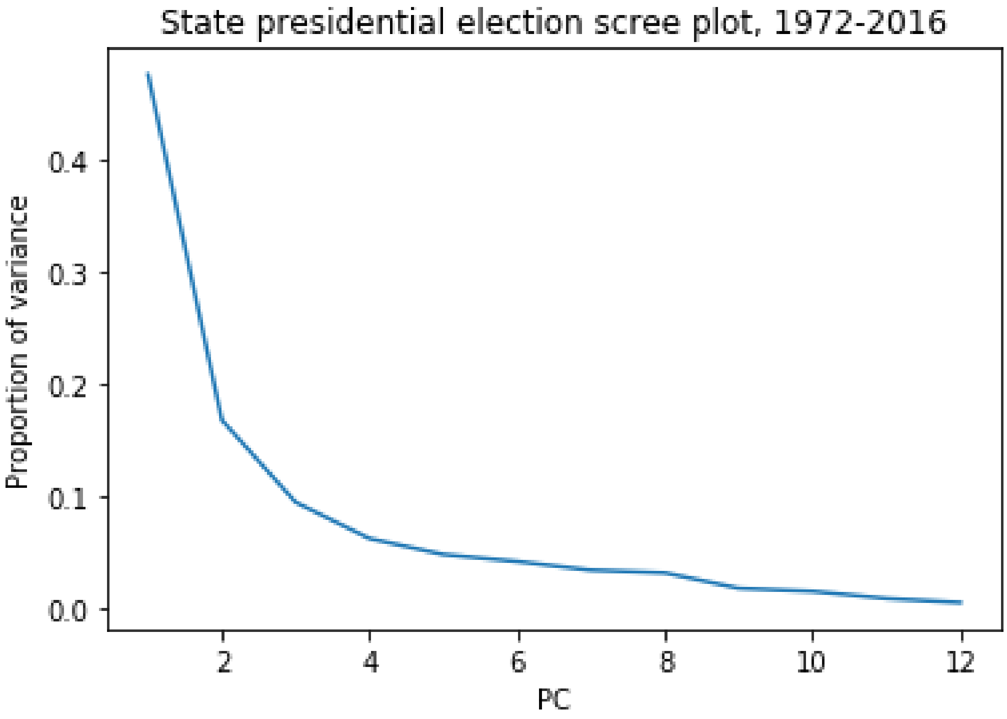
## 0.4 Question 3j

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 [185]: pc_i = np.arange(1, len(s_q3)+1)
          (sns.lineplot(x=pc_i, y=s_q3**2 / sum(s_q3**2))
              .set(
                  title="State presidential election scree plot, 1972-2016",
                  xlabel="PC",
                  ylabel="Proportion of variance"
              )
          )
```

```
Out[185]: [Text(0, 0.5, 'Proportion of variance'),
           Text(0.5, 0, 'PC'),
           Text(0.5, 1.0, 'State presidential election scree plot, 1972-2016')]
```


State presidential election scree plot, 1972-2016
Proportion of variance
PC