0.0.1 Question 1a

What does each row in `df_clean` represent?

Each row represents a state in the U.S. svoted in presidential elections between 1972 and 2016.

Unfortunately, we have two problems:

1. There is a lot of overplotting, with only 28 distinct dots (out of 104 points). 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.

0.0.2 Question 2a: Jitter

Let's start by addressing problem 1.

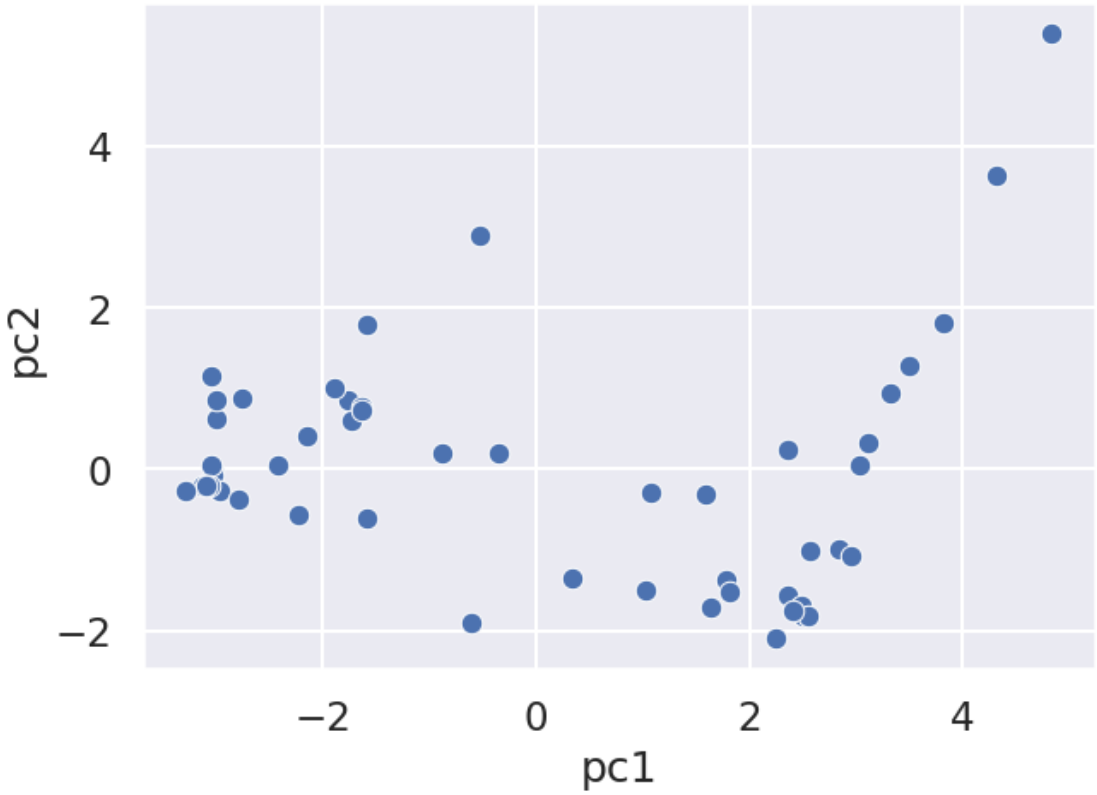
In the cell below, create a new dataframe `first_2_pcs_jittered` with a small amount of random noise added to each principal component. In this same cell, create a scatterplot.

To reduce overplotting, we **jitter** the first two principal components: * Add a small amount of random, unbiased Gaussian noise to each value using `np.random.normal` (documentation) with mean 0 and standard deviation less than 1. * 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, but without overplotting. * The amount of noise you add *should not significantly affect* the appearance of the plot; it should simply serve to separate overlapping observations.

In [ ]:

In [57]:
```
# first, jitter the data
first_2_pcs_jittered = pd.DataFrame({
    "pc1": first_2_pcs["pc1"] + np.random.normal(loc = 0, scale = 0.15, size = first_2_pcs.sha
    "pc2": first_2_pcs["pc2"] + np.random.normal(loc = 0, scale = 0.15, size = first_2_pcs.sha
})

# then, create a scatter plot
sns.scatterplot(data=first_2_pcs_jittered, x="pc1", y="pc2");
```


pc2
4
2
0
−2
−2
0
2
4
pc1

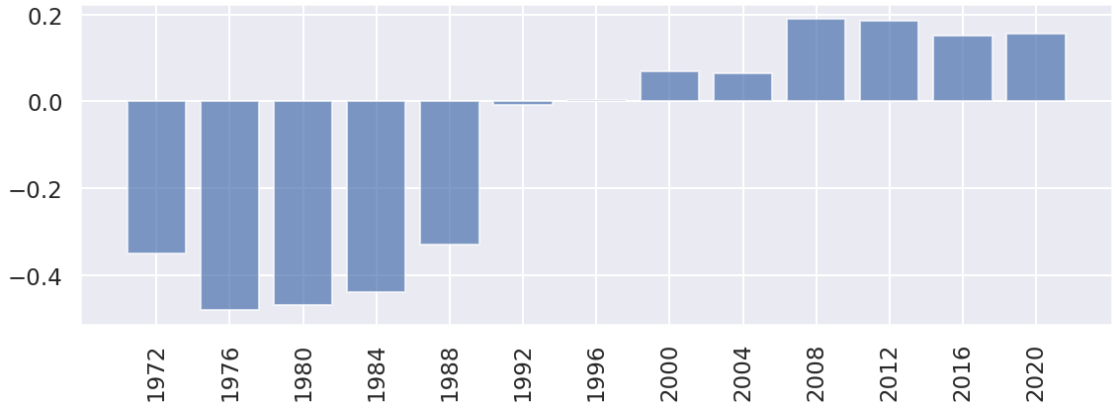
Analyze the above plot. In the below cell, address the following two points: 1. Give an example of a cluster of states that vote a similar way. Does the composition of this cluster surprise you? If you're not familiar with U.S. politics, it's fine to just say "No, I'm not surprised because I don't know anything about U.S. politics." 1. Include anything interesting that you observe. You will get credit for this as long as you write something reasonable that you can takeaway from the plot.

One example of cluster of state that vote in a similar way is close to (-3, 0) like Kansas, Alaska, Wyoming, and some other. Looking at this cluster and some others (like states close to (2, -2)) surprised me because I didn't know states votes similarly for a particular party. The plot also show party line which I found important and interesting. Besides, I noticed that two state have much higher pc1 and pc2 values like Washington D.C. and Minnesota, which I assume is due to the variance in voting.

0.0.3 Question 3a

In the cell below, plot the the 2nd row of V^T, i.e., the row of V^T that correpsonds to `pc2`.

```
In [60]: plt.figure(figsize=(12, 4))
         plot_pc(list(df_standardized.columns), vt, 1);
```



0.0.4 Question 3b

Using the two above plots of the rows of V^T as well as the original table, give a description of what it means for a point to be in the top-right quadrant of the 2-D scatter plot from Question 2.

In other words, what is generally true about a state with relatively large positive value for `pc1` (right side of 2-D scatter plot)? For a large positive value for `pc2` (top side of 2-D scatter plot)?

Notes: * `pc2` is pretty hard to interpret, and the staff doesn't really have a consensus on what it means either. We'll be very nice when grading as long as your answer is reasonable - there is no correct answer necessarily. * Principal components beyond the first are often hard to interpret (but not always; see the lab).

As I stated earlier pc1 describes party line. I believe state with large positive vlaue of pc1 are democrats and vice versa. States with large positive value of pc2 shows variance in vote, however, I'm 100% sure.

```
In [61]: # feel free to use this cell for scratch work. If you need more scratch space, add cells *belo

         # Make sure to put your actual answer in the cell above where it says "Type your answer here,
```

0.0.5 Question 3c

To get a better sense of whether our 2D scatterplot captures the whole story, create a **scree plot** for this data. In other words, plot the fraction of the total variance (y-axis) captured by the ith principal component (x-axis).

Hint: Be sure to label your axes appropriately! You may find `plt.xticks()` (documentation) helpful for formatting. Also check out the lab for more on scree plots.

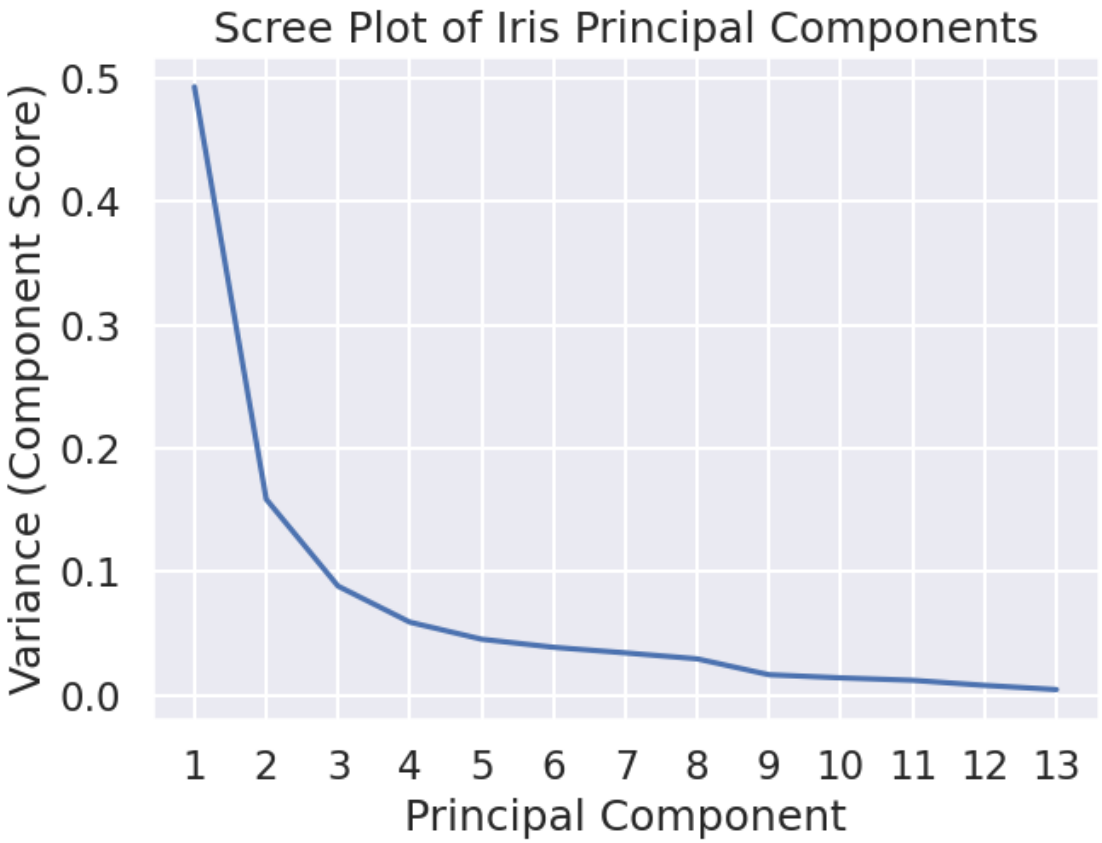
```
In [72]: list(range(1, 13))
```

```
Out[72]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
In [25]: x_range = list(range(1, 14))
         plt.xticks(x_range, x_range)
         plt.plot(x_range, s**2 / np.sum(s**2))

         plt.title('Scree Plot of Iris Principal Components')
         plt.xlabel('Principal Component')
         plt.ylabel('Variance (Component Score)')
```

```
Out[25]: Text(0, 0.5, 'Variance (Component Score)')
```


Scree Plot of Iris Principal Components
Variance (Component Score)
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Principal Component