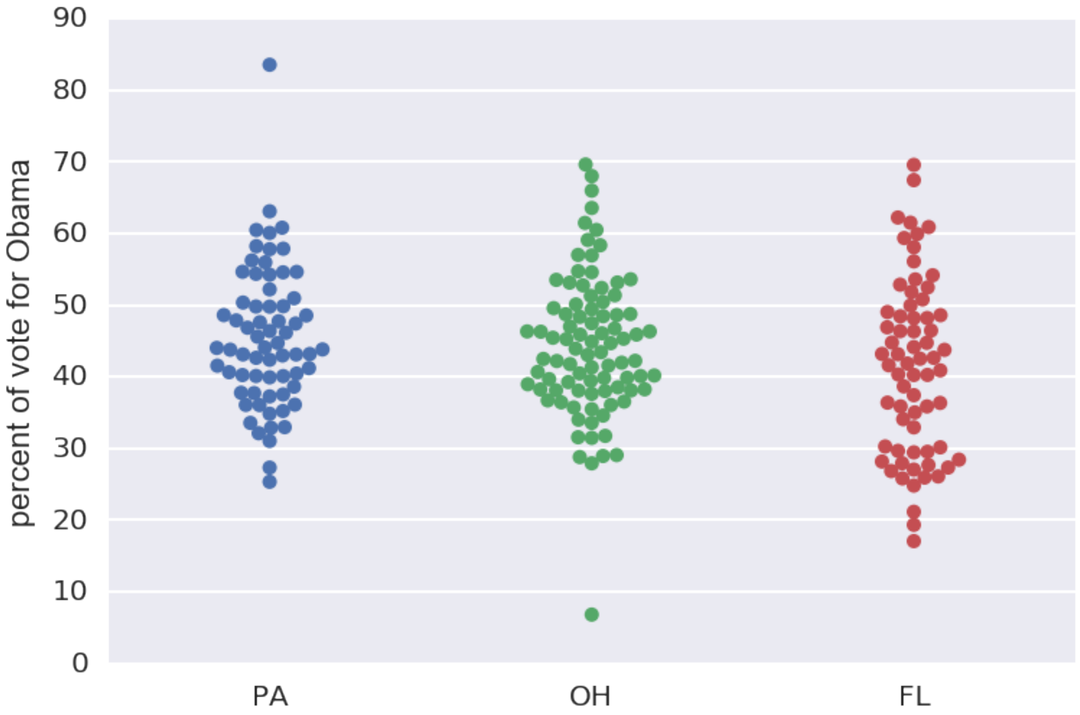
STATISTICAL THINKING IN PYTHON I

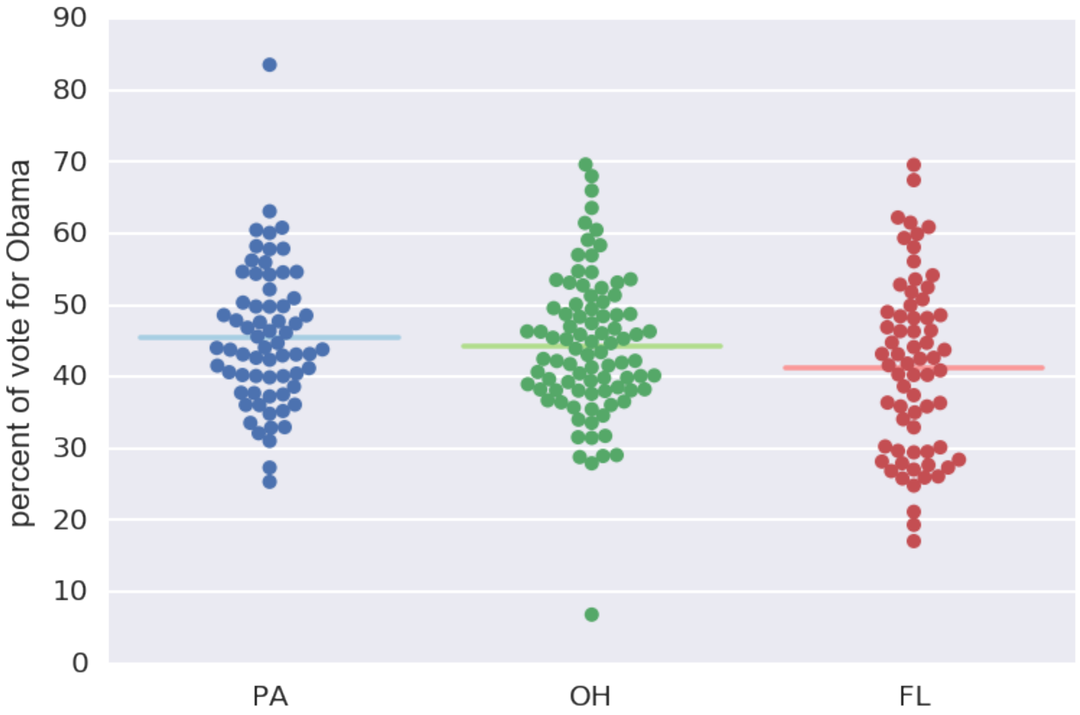
Introduction to summary statistics: The sample mean and median

2008 US swing state election results



Data retrieved from Data.gov (https://www.data.gov/)

2008 US swing state election results



Data retrieved from Data.gov (https://www.data.gov/)

Mean vote percentage

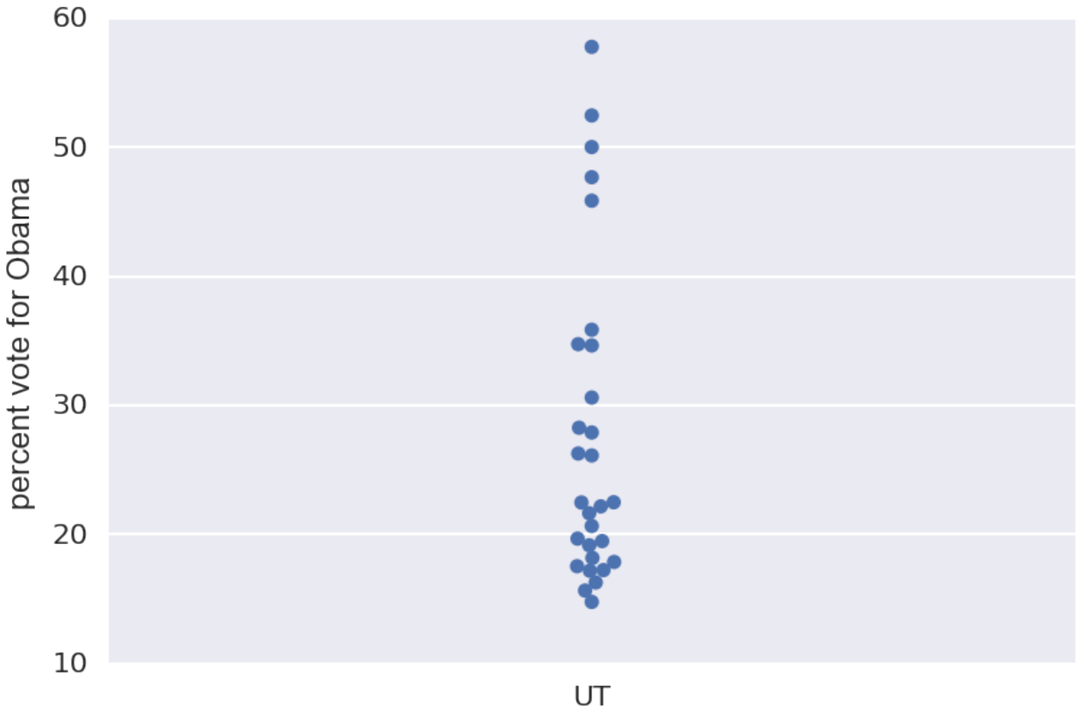
```
In [1]: import numpy as np

In [2]: np.mean(dem_share_PA)
Out[2]: 45.476417910447765
```

$$\text{mean} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Outliers

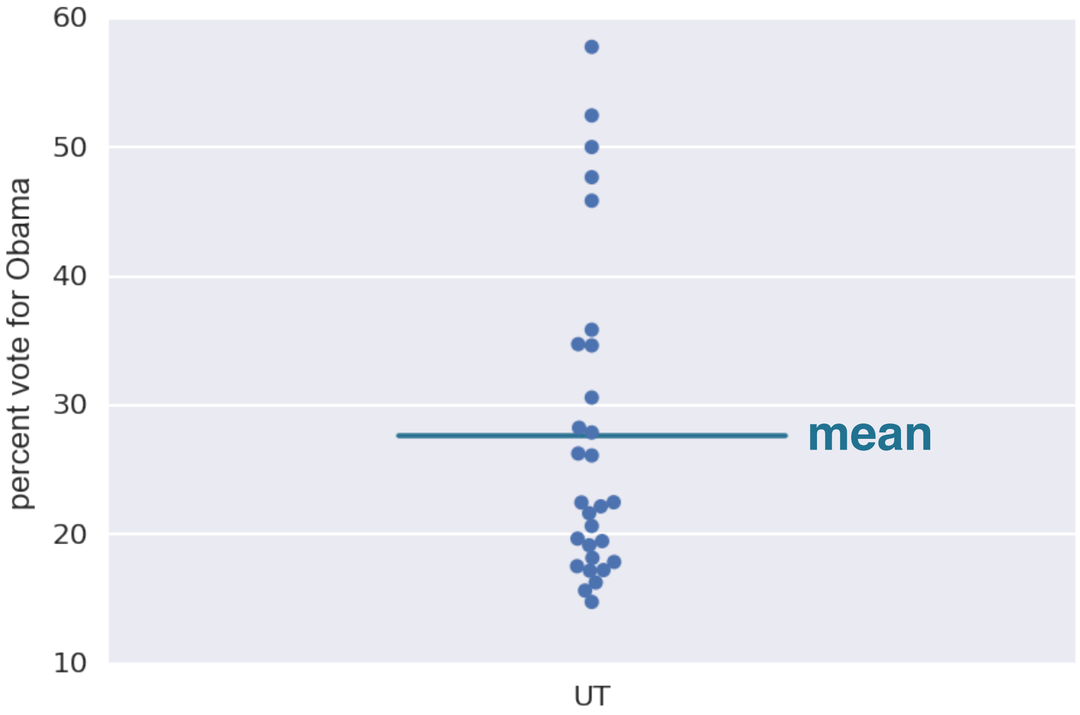
- Data points whose value is far greater or less than most of the rest of the data

2008 Utah election results



Data retrieved from Data.gov (https://www.data.gov/)

2008 Utah election results

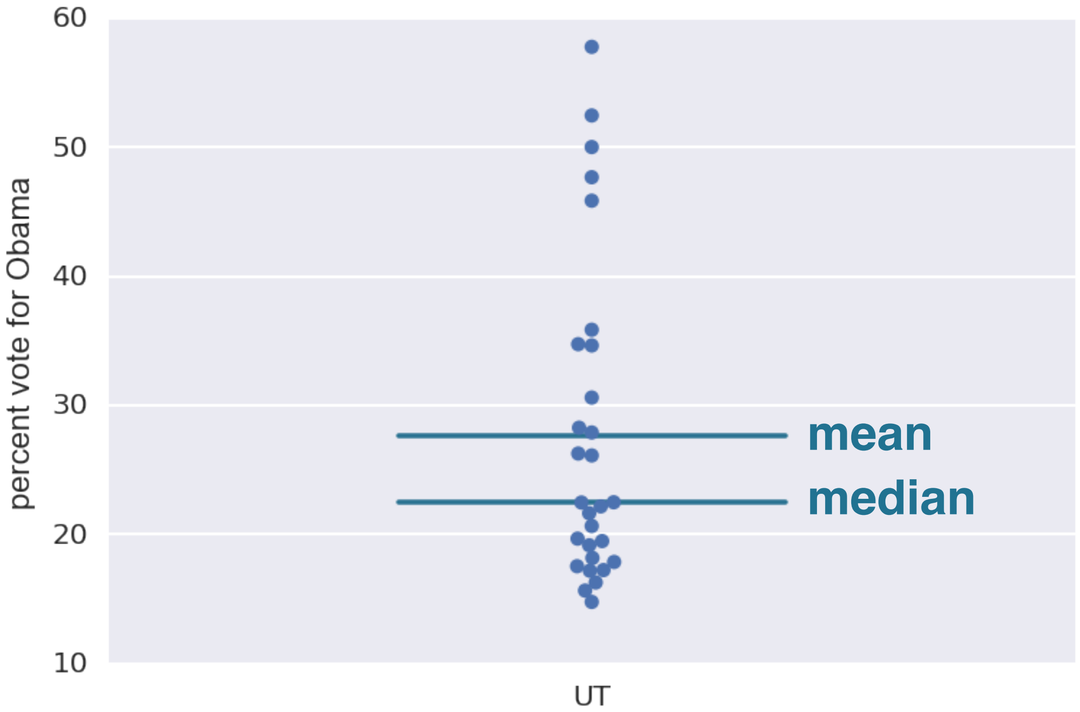


Data retrieved from Data.gov (https://www.data.gov/)

The median

- The middle value of a data set

2008 Utah election results



Data retrieved from Data.gov (https://www.data.gov/)

Computing the median

```
In [1]: np.median(dem_share_UT)
Out[1]: 22.469999999999999
```

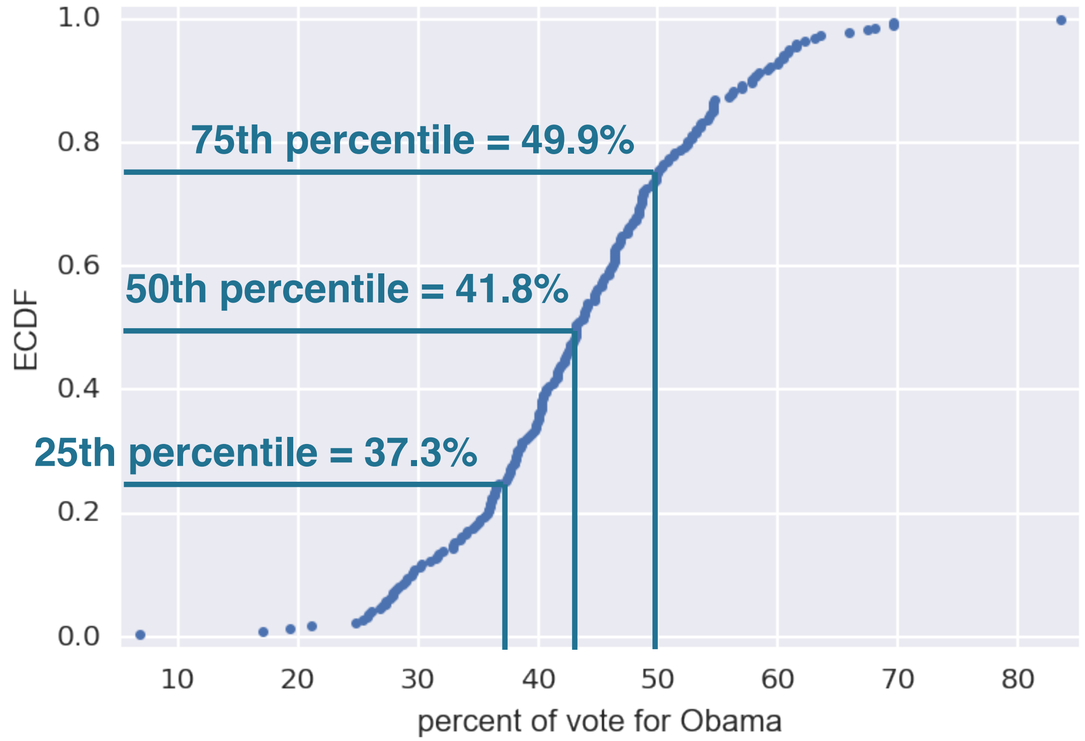
STATISTICAL THINKING IN PYTHON I

Let’s practice!

STATISTICAL THINKING IN PYTHON I

Percentiles, outliers, and box plots

Percentiles on an ECDF

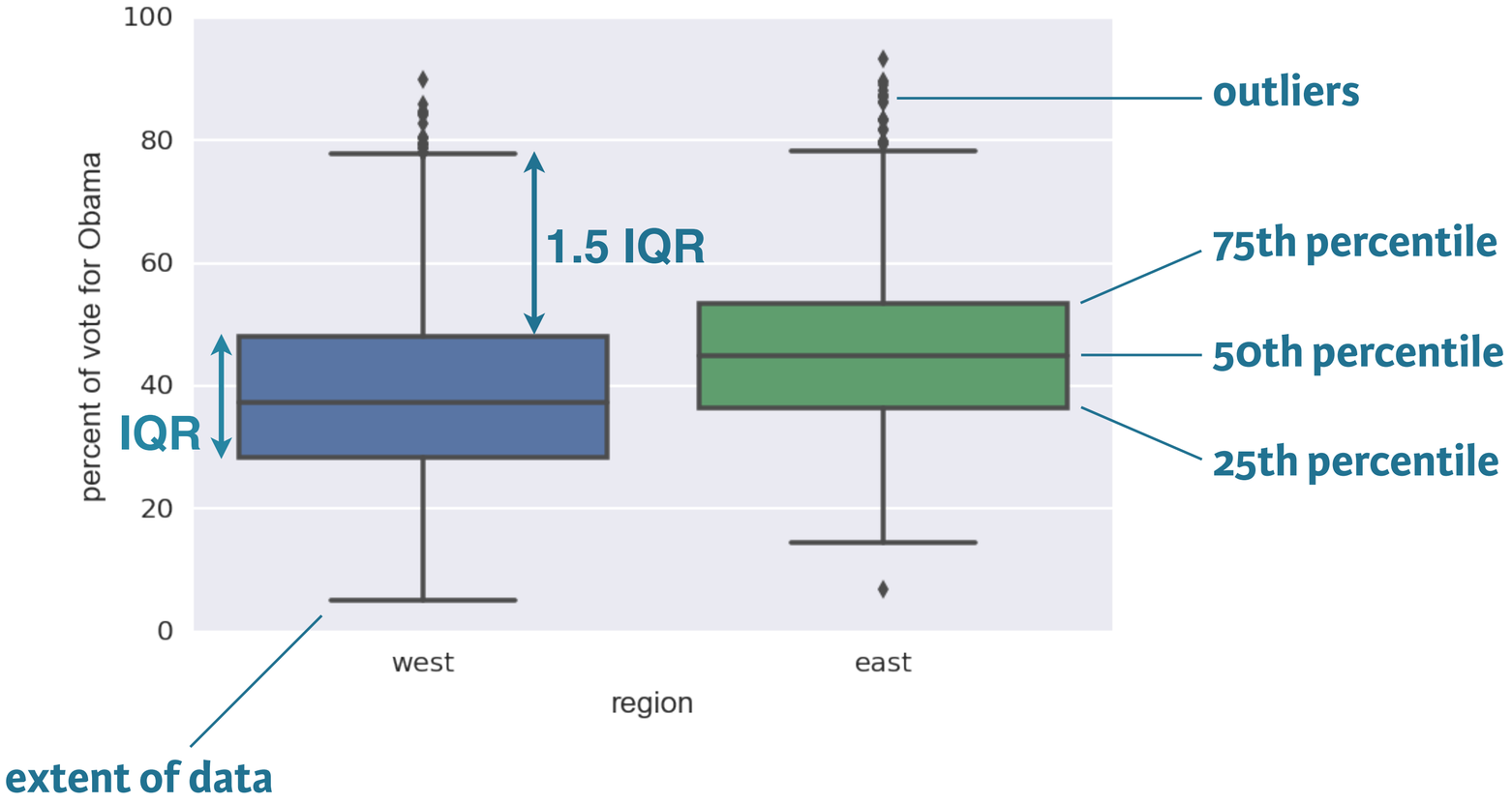


Data retrieved from Data.gov (https://www.data.gov/)

Computing percentiles

```
In [1]: np.percentile(df_swing['dem_share'], [25, 50, 75])
Out[1]: array([ 37.3025,  43.185 ,  49.925 ])
```

2008 US election box plot



Data retrieved from Data.gov (https://www.data.gov/)

Generating a box plot

```
In [1]: import matplotlib.pyplot as plt

In [2]: import seaborn as sns

In [3]: _ = sns.boxplot(x='east_west', y='dem_share',
   ...:                 data=df_all_states)

In [4]: _ = plt.xlabel('region')

In [5]: _ = plt.ylabel('percent of vote for Obama')

In [6]: plt.show()
```

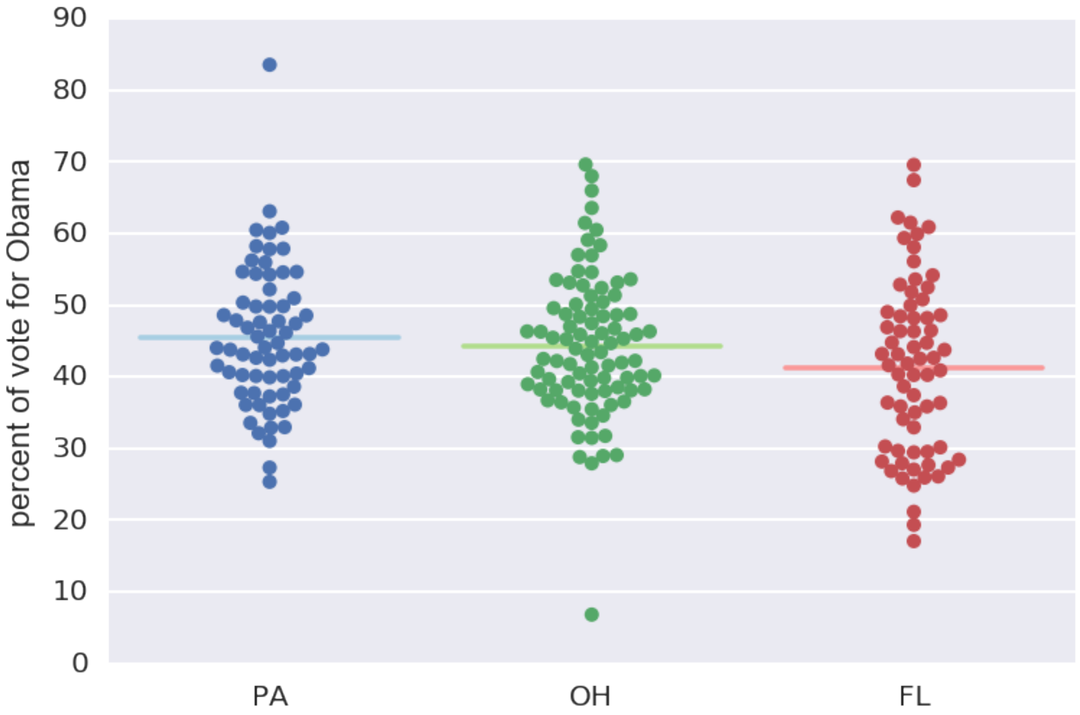
STATISTICAL THINKING IN PYTHON I

Let’s practice!

STATISTICAL THINKING IN PYTHON I

Variance and standard deviation

2008 US swing state election results

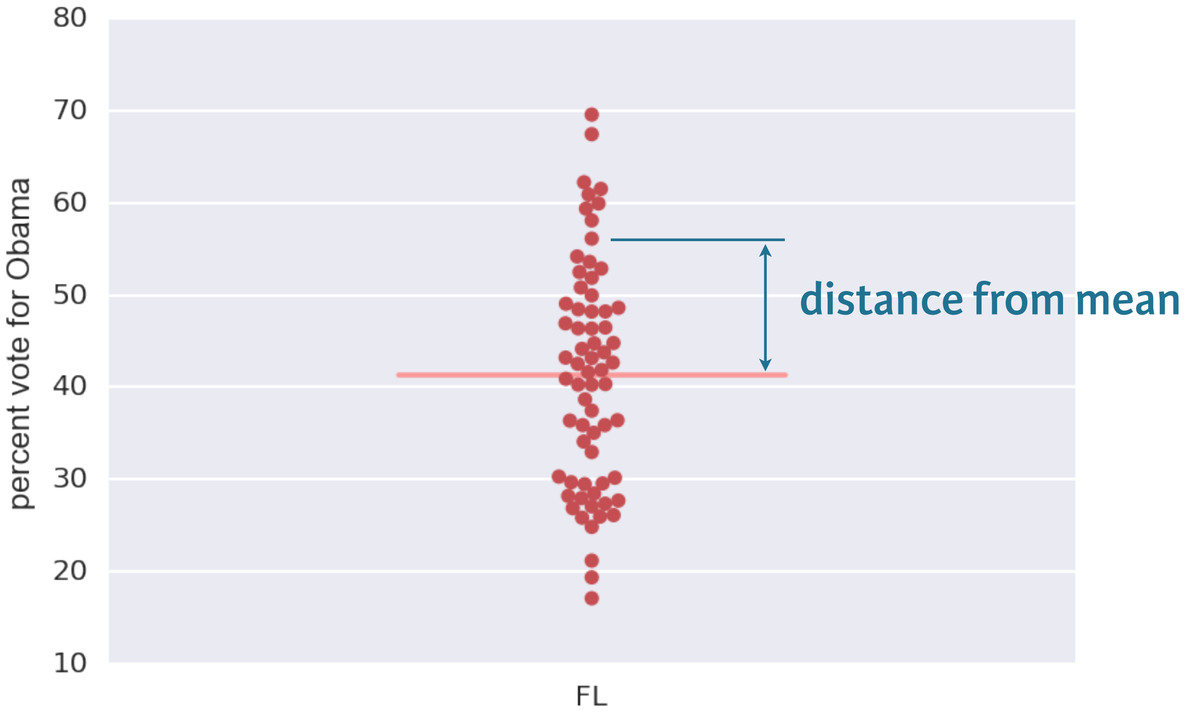


Data retrieved from Data.gov (https://www.data.gov/)

Variance

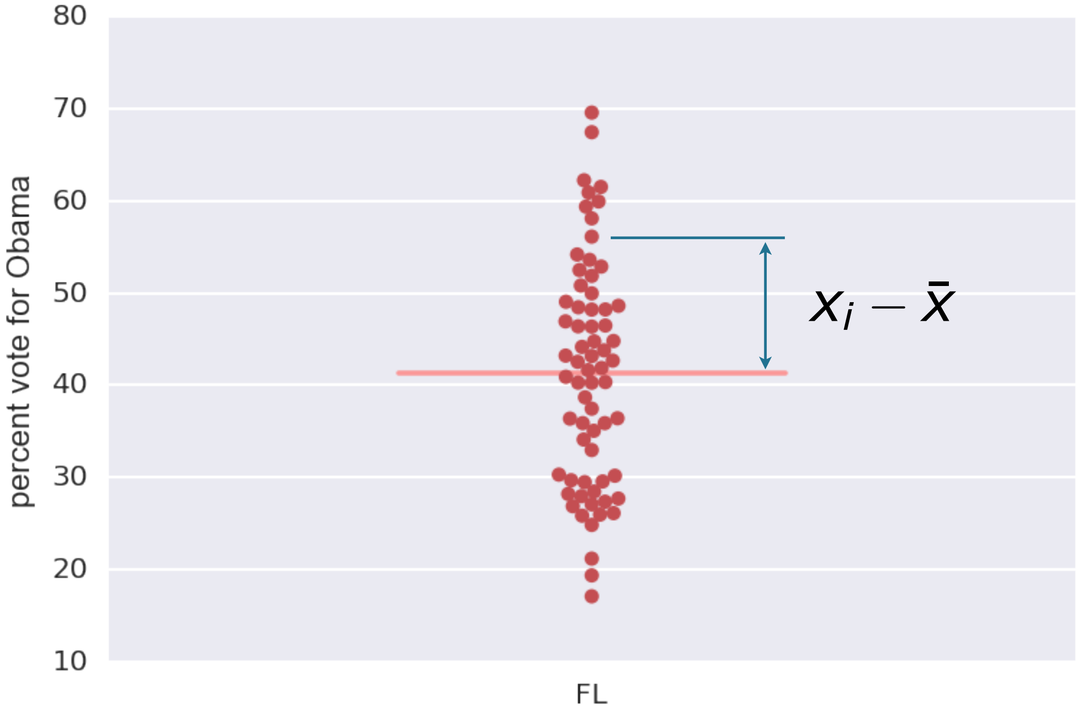
- The mean squared distance of the data from their mean
- Informally, a measure of the spread of data

2008 Florida election results



Data retrieved from Data.gov (https://www.data.gov/)

2008 Florida election results



$$\text{variance} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Data retrieved from Data.gov (https://www.data.gov/)

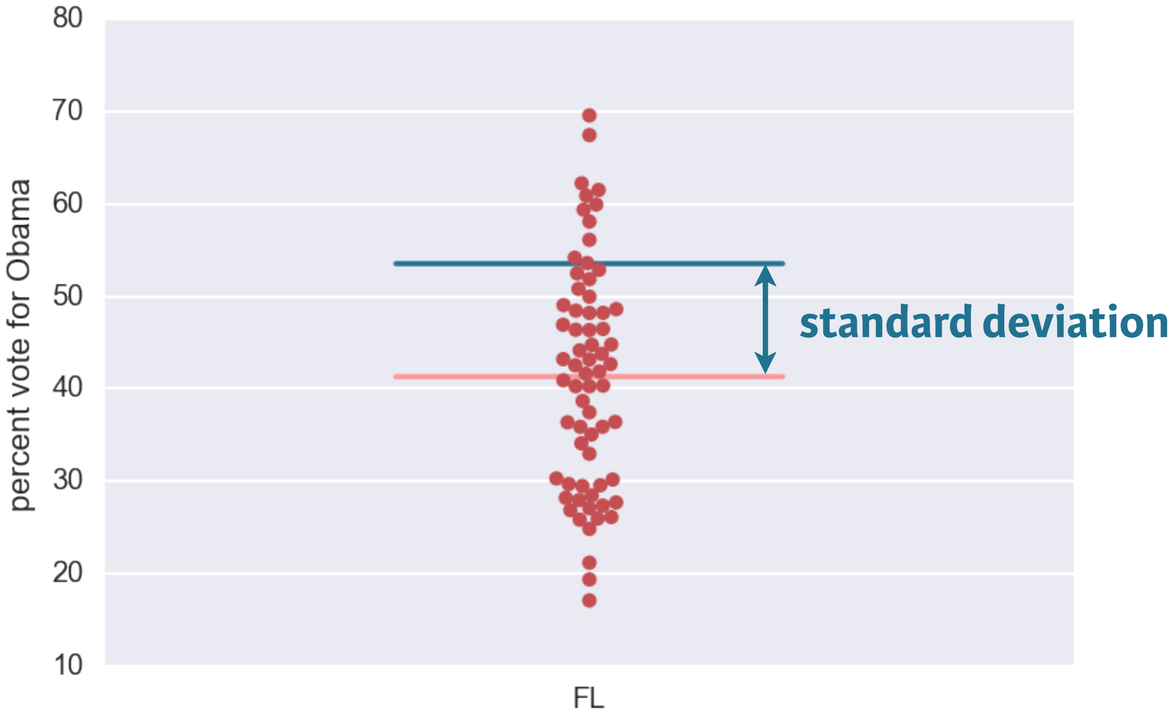
Computing the variance

```
In [1]: np.var(dem_share_FL)
Out[1]: 147.44278618846064
```

Computing the standard deviation

```
In [1]: np.std(dem_share_FL)
Out[1]: 12.142602117687158

In [2]: np.sqrt(np.var(dem_share_FL))
Out[2]: 12.142602117687158
```

2008 Florida election results



Data retrieved from Data.gov (https://www.data.gov/)

STATISTICAL THINKING IN PYTHON I

Let’s practice!

STATISTICAL THINKING IN PYTHON I

Covariance and the Pearson correlation coefficient

2008 US swing state election results

Data retrieved from Data.gov (https://www.data.gov/)

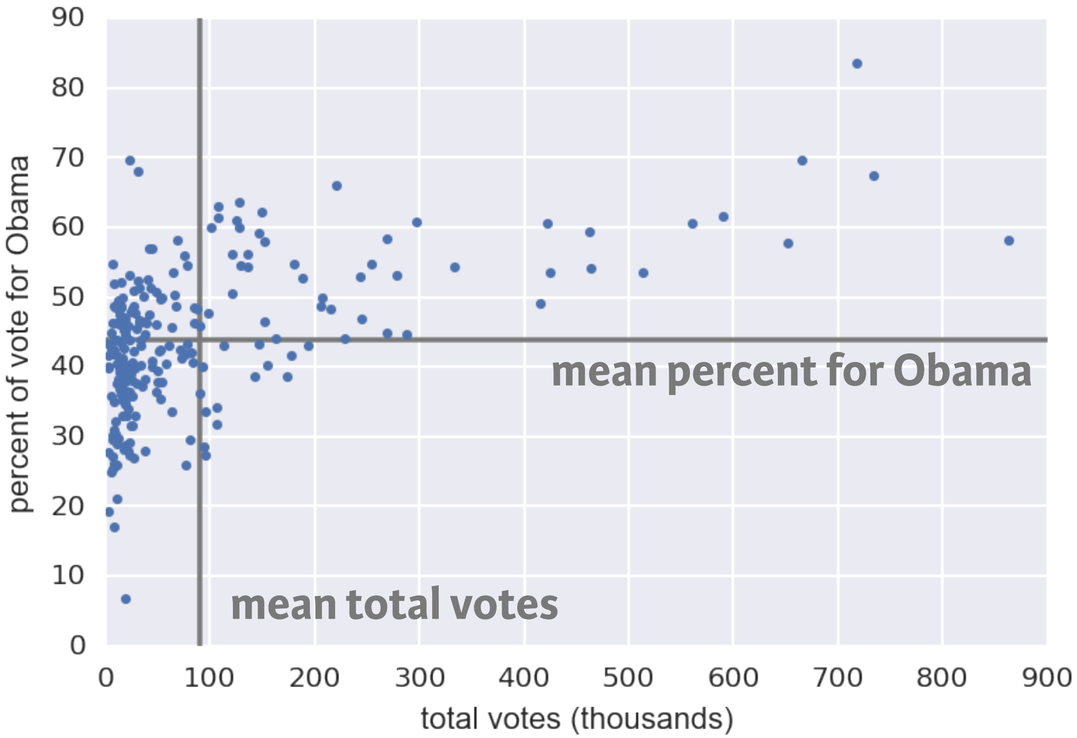
Generating a scatter plot

```
In [1]: _ = plt.plot(total_votes/1000, dem_share,
   ...:                marker='.', linestyle='none')

In [2]: _ = plt.xlabel('total votes (thousands)')

In [3]: _ = plt.ylabel('percent of vote for Obama')
```

Covariance

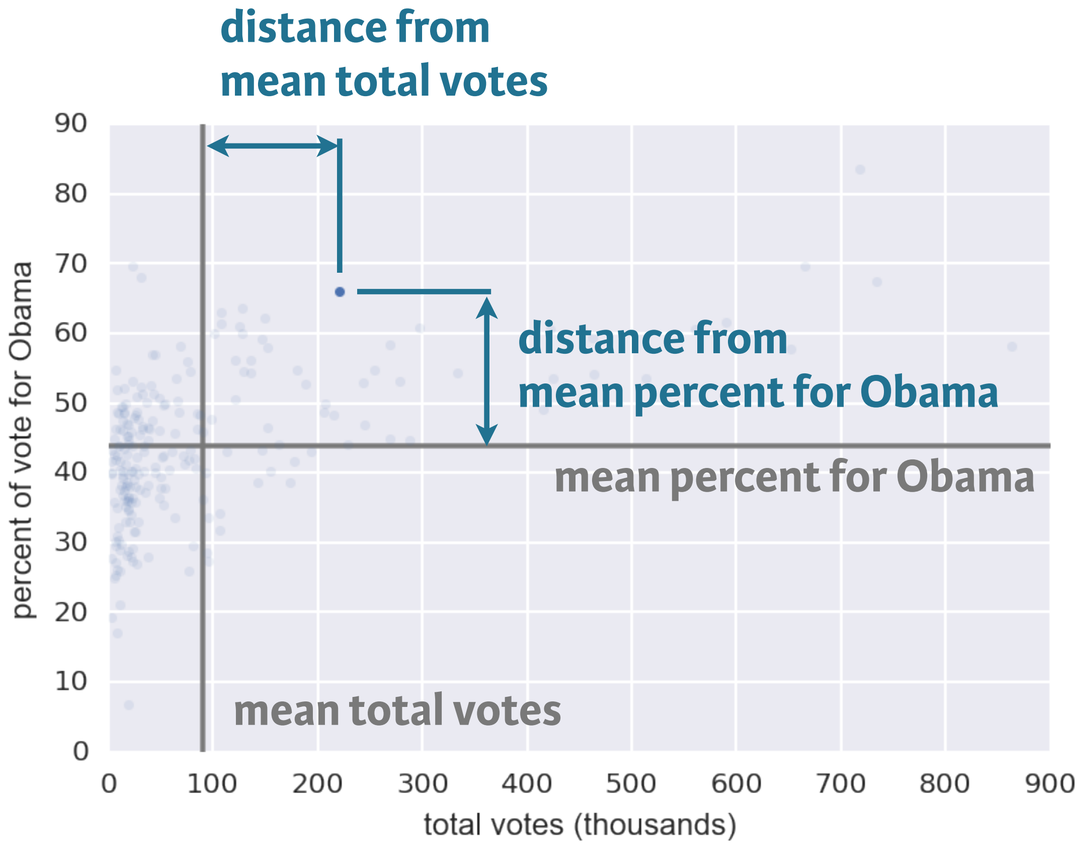
- A measure of how two quantities vary *together*

Calculation of the covariance



Data retrieved from Data.gov (https://www.data.gov/)

Calculation of the covariance



$$\text{covariance} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

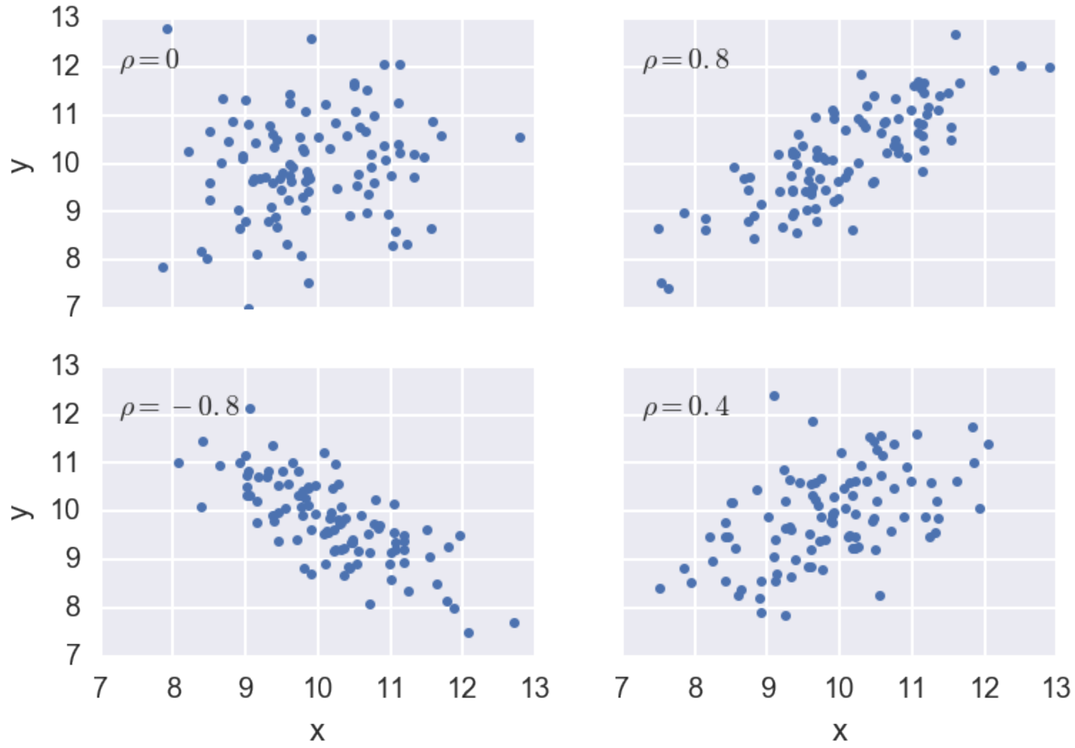
Data retrieved from Data.gov (https://www.data.gov/)

Pearson correlation coefficient

$$\rho = \text{Pearson correlation} = \frac{\text{covariance}}{(\text{std of } x)\,(\text{std of } y)}$$

$$= \frac{\text{variability due to codependence}}{\text{independent variability}}$$

Pearson correlation coefficient examples



STATISTICAL THINKING IN PYTHON I

Let’s practice!