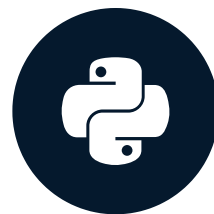


# Introduction to summary statistics: The sample mean and median

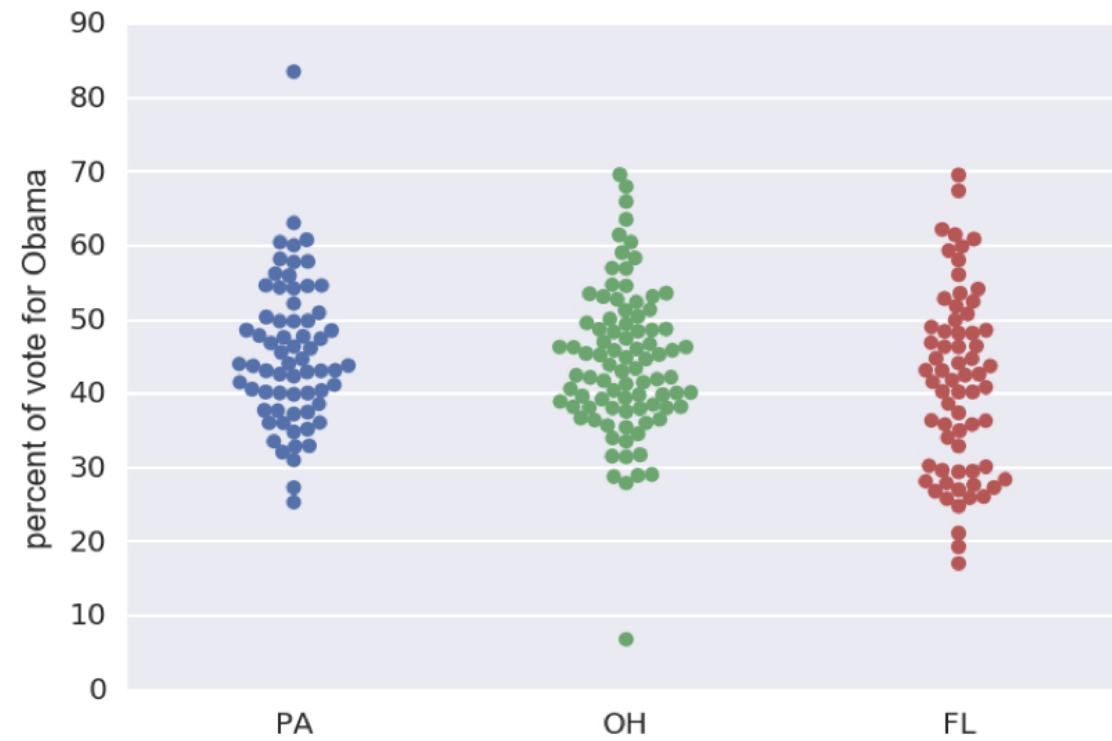
STATISTICAL THINKING IN PYTHON (PART 1)

**Justin Bois**

Lecturer at the California Institute of  
Technology

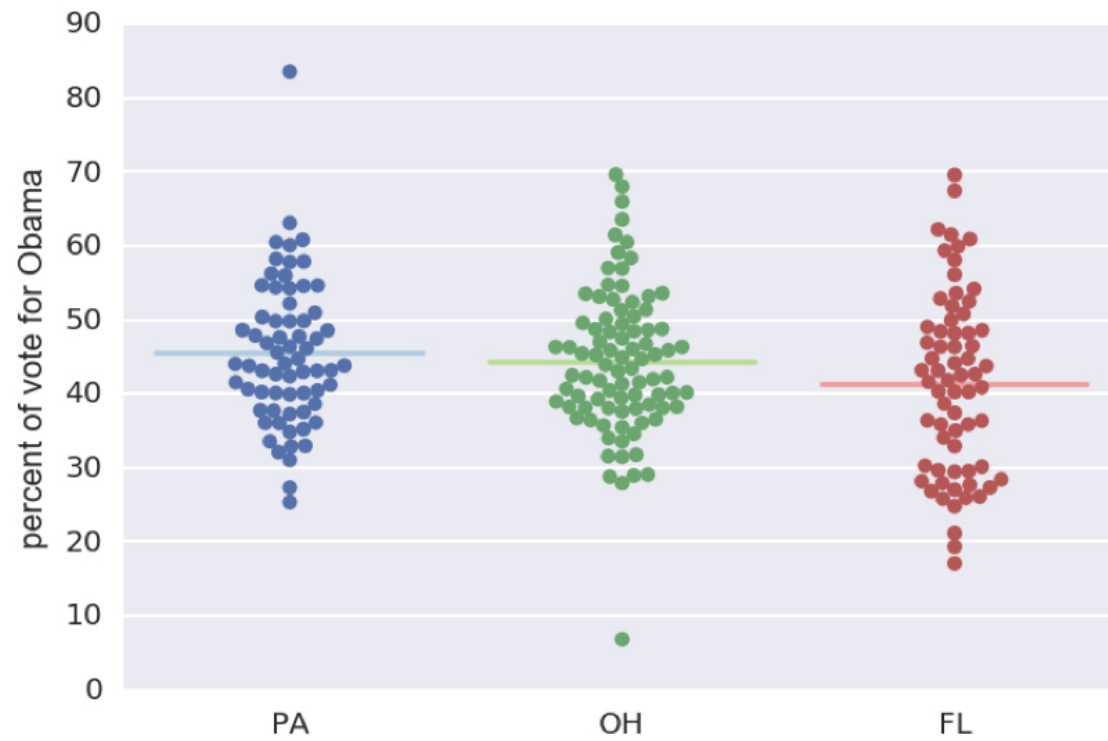


# 2008 US swing state election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Mean vote percentage

```
import numpy as np  
np.mean(dem_share_PA)
```

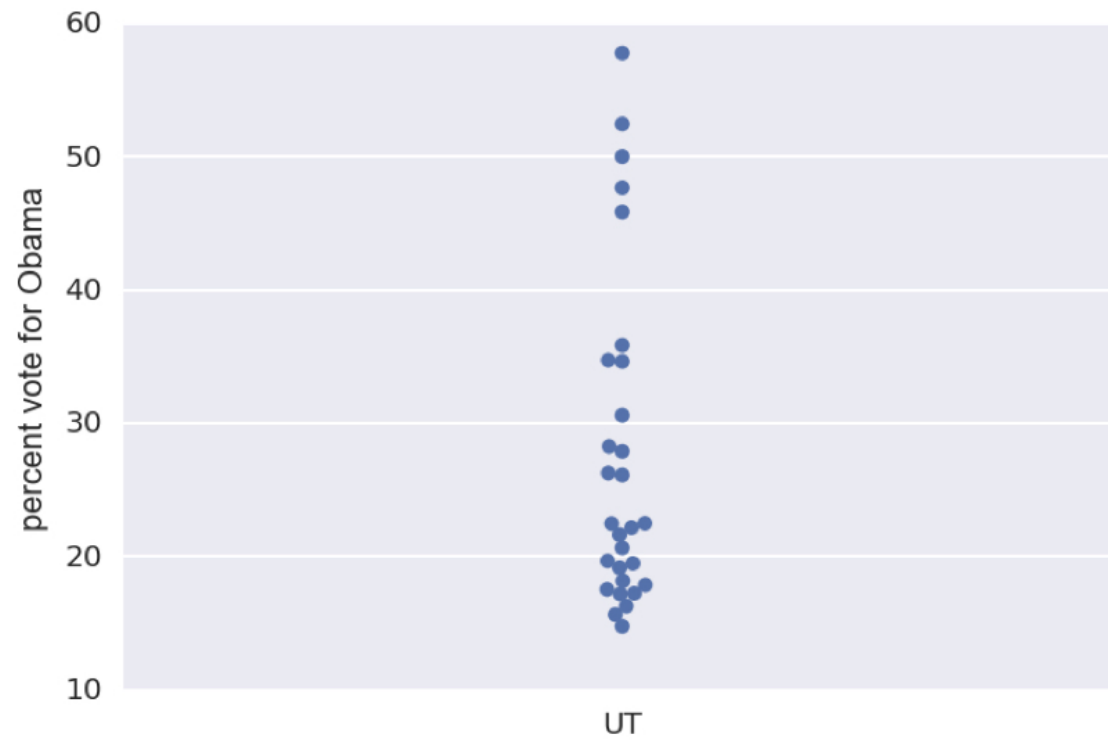
```
45.476417910447765
```

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

# Outliers

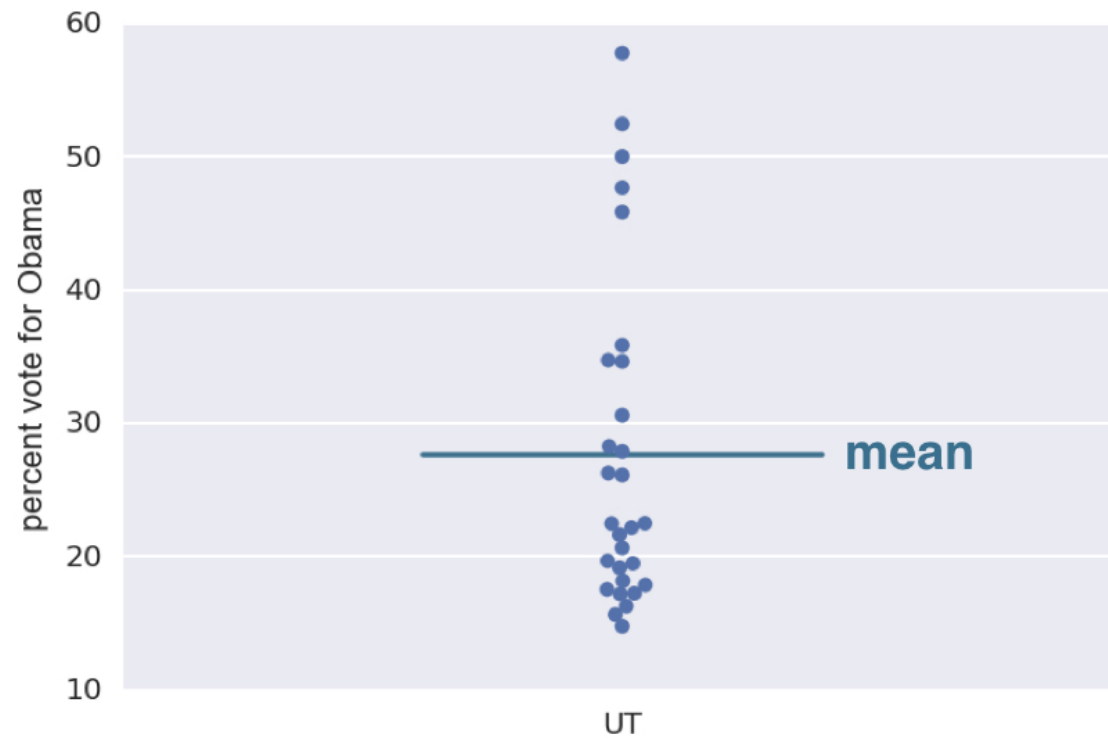
- Data points whose value is far greater or less than most of the rest of the data

# 2008 Utah election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 Utah election results



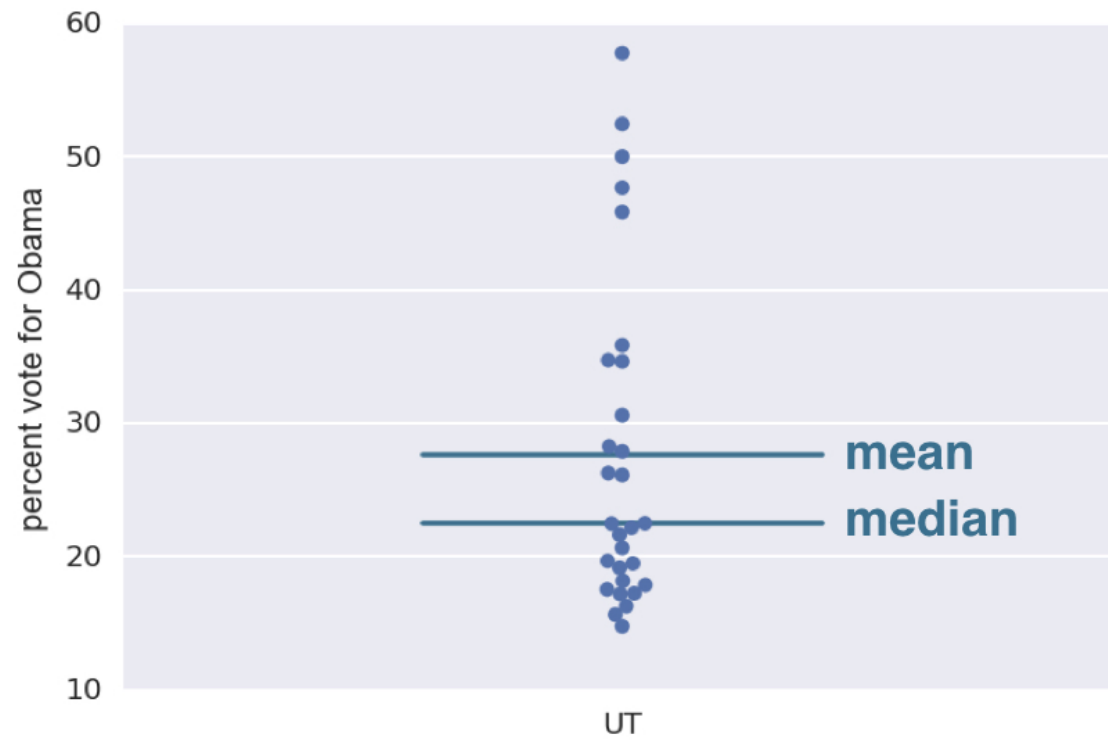
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# The median

- The middle value of a data set



# 2008 Utah election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Computing the median

```
np.median(dem_share_UT)
```

```
22.469999999999999
```

# Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

# Percentiles, outliers, and box plots

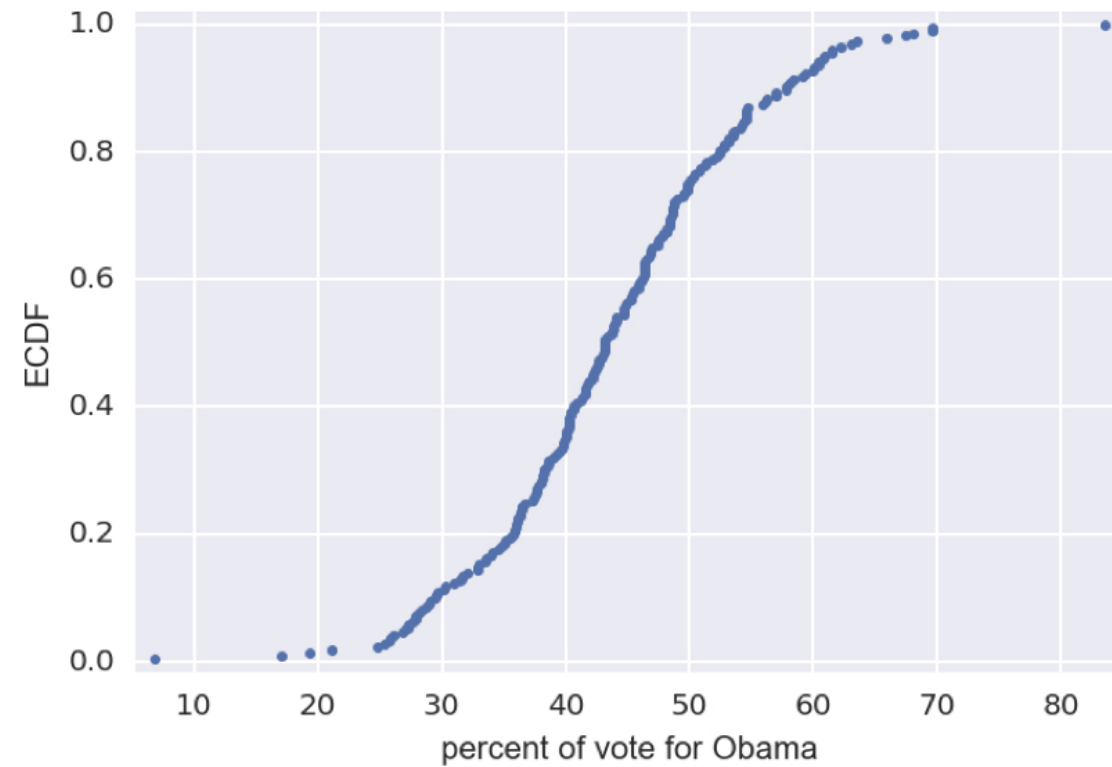
STATISTICAL THINKING IN PYTHON (PART 1)



**Justin Bois**

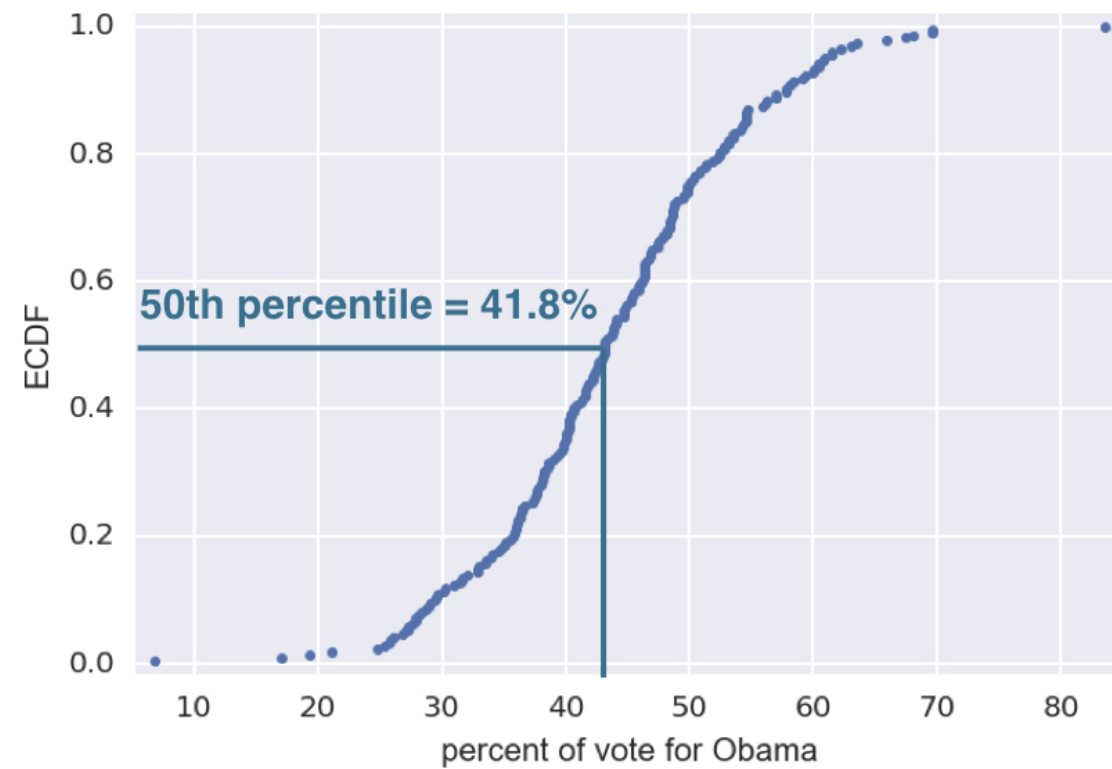
Lecturer at the California Institute of  
Technology

# Percentiles on an ECDF

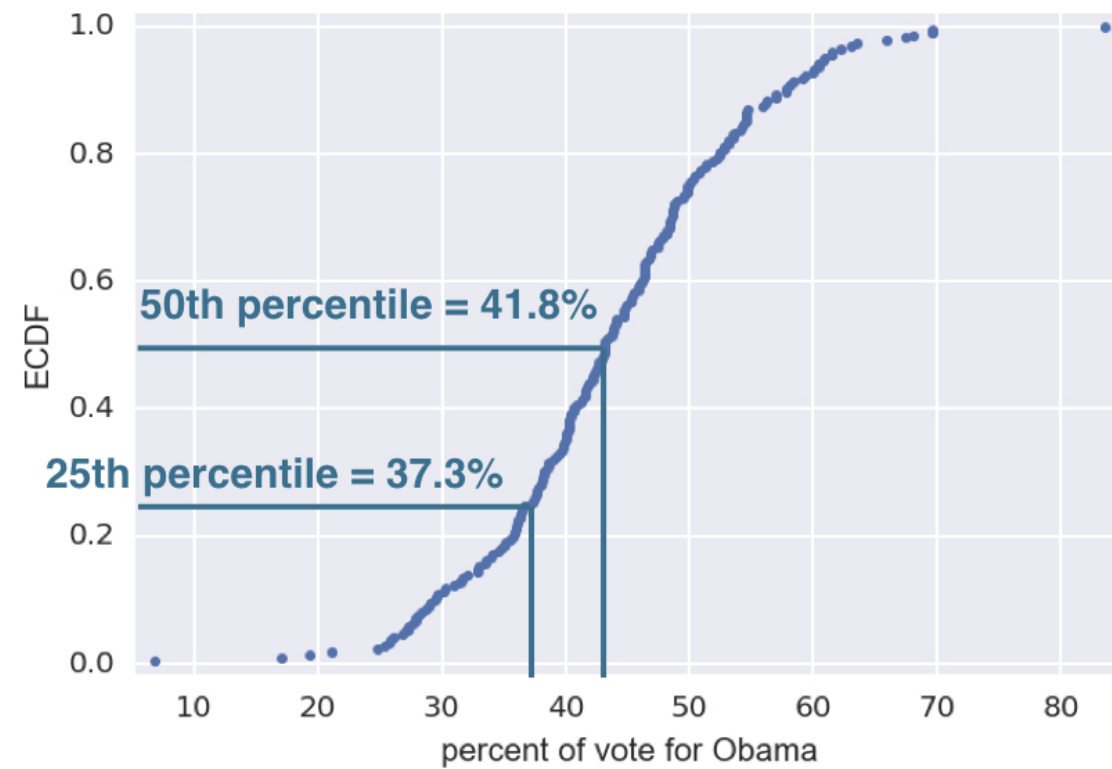


<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

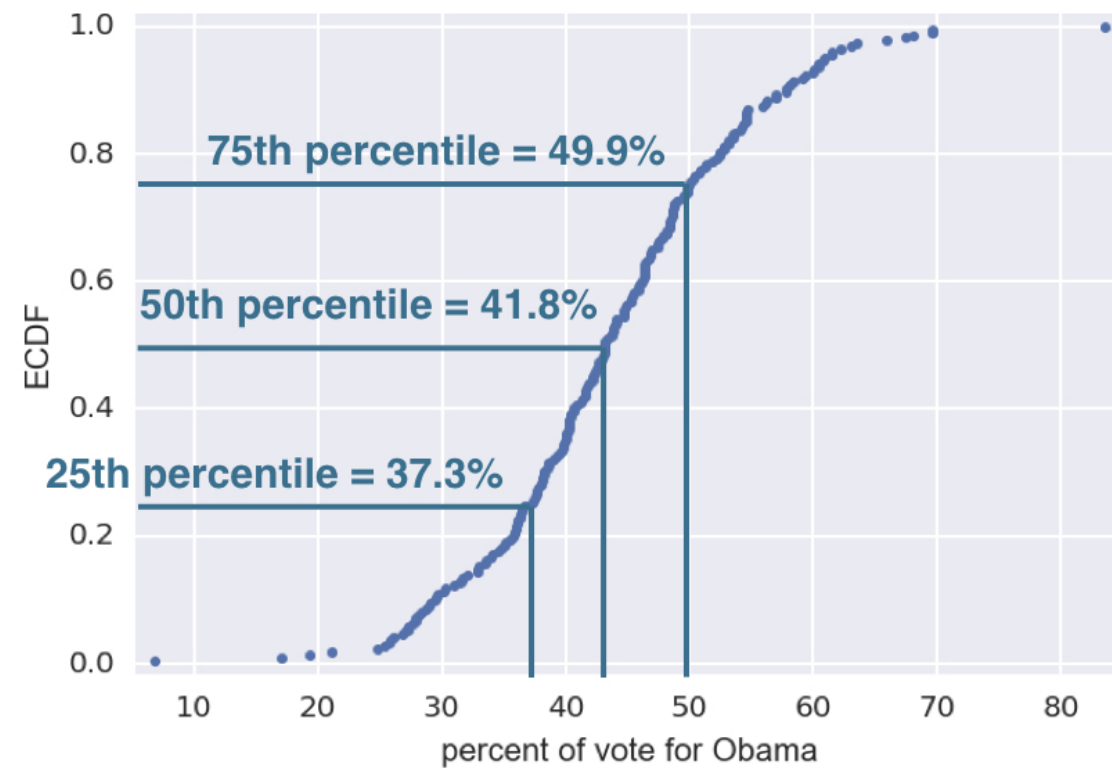
# Percentiles on an ECDF



# Percentiles on an ECDF



# Percentiles on an ECDF



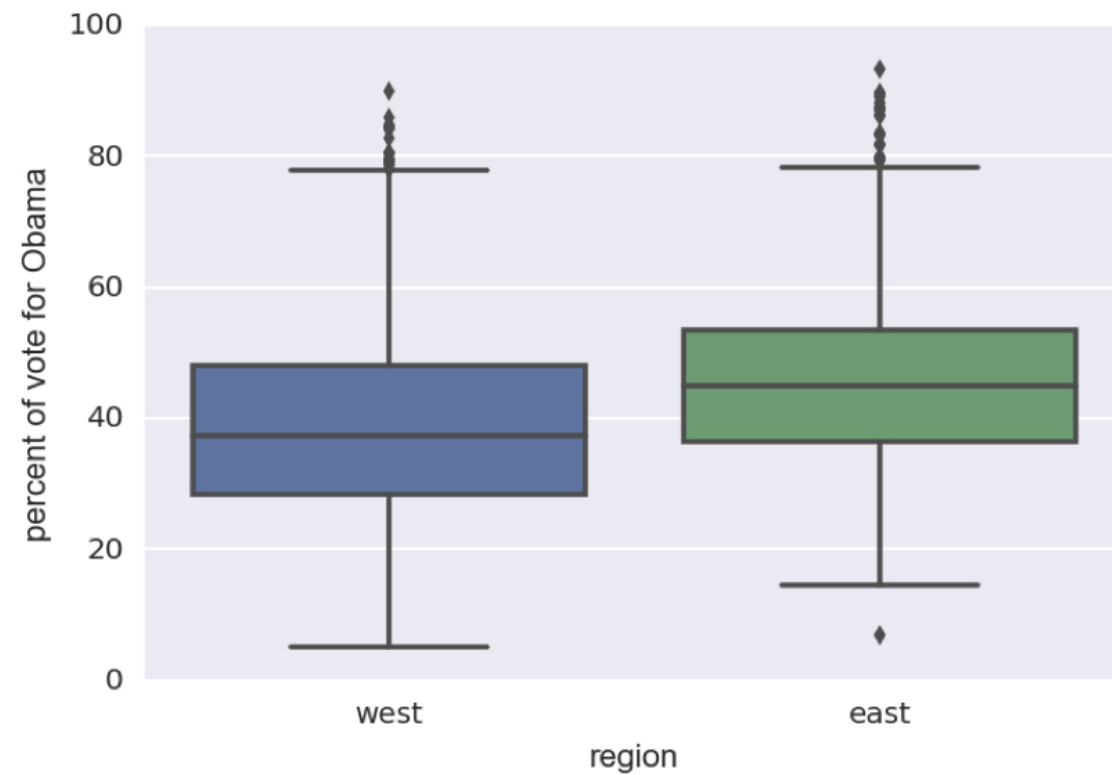


# Computing percentiles

```
np.percentile(df_swing['dem_share'], [25, 50, 75])
```

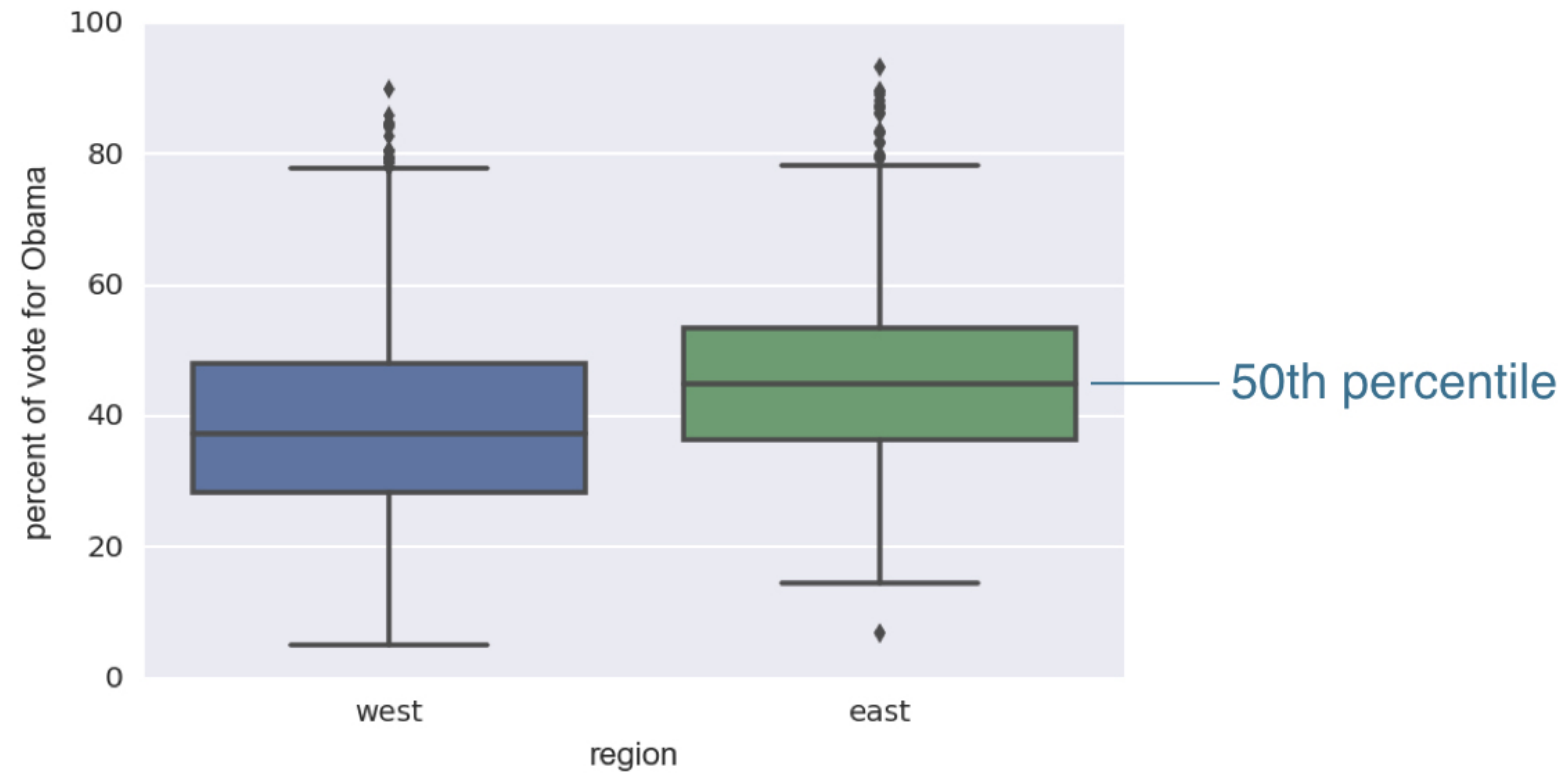
```
array([ 37.3025,  43.185 ,  49.925 ])
```

# 2008 US election box plot

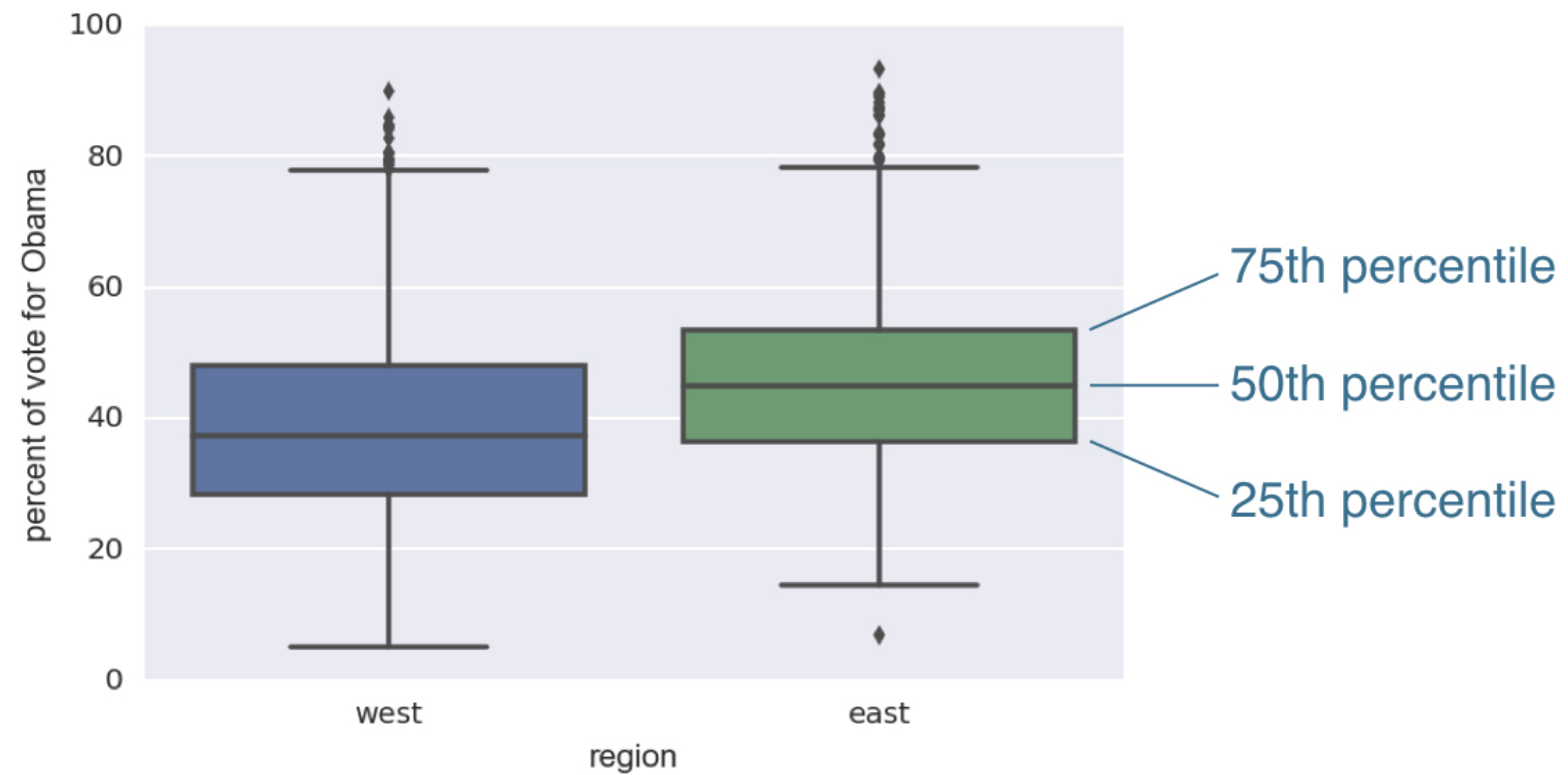


<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

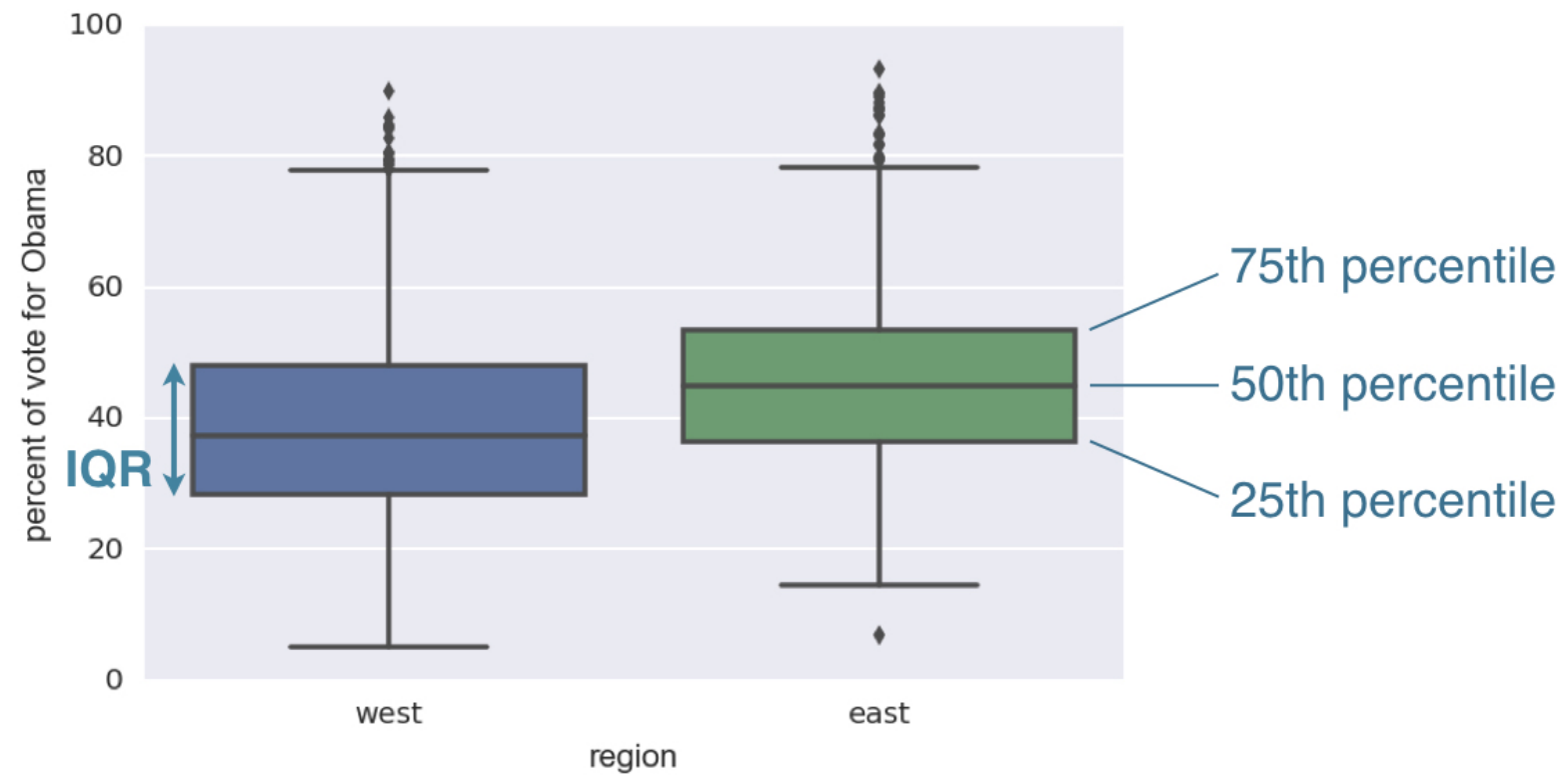
# 2008 US election box plot



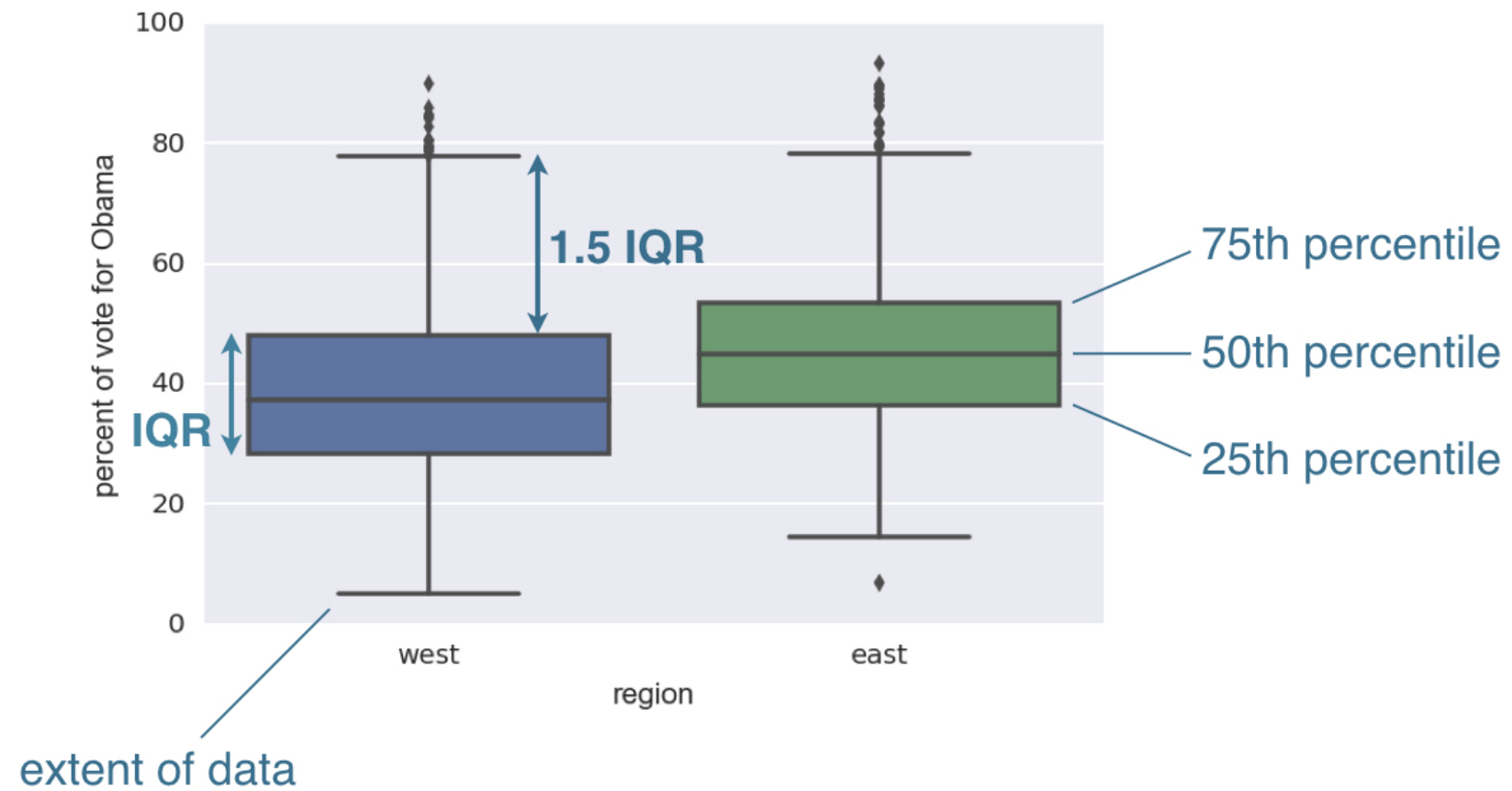
# 2008 US election box plot



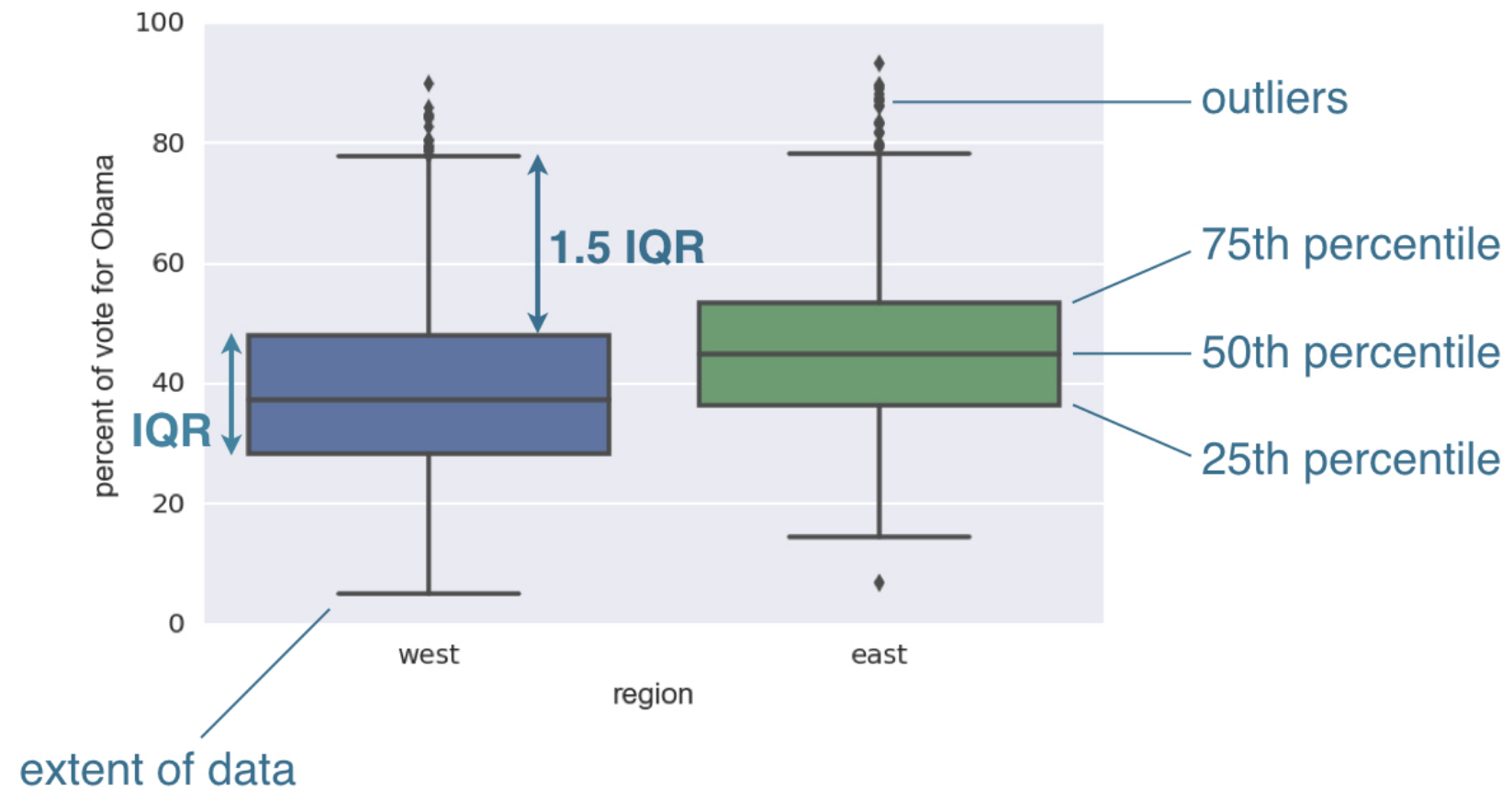
# 2008 US election box plot



# 2008 US election box plot



# 2008 US election box plot



# Generating a box plot

```
import matplotlib.pyplot as plt
import seaborn as sns
_ = sns.boxplot(x='east_west', y='dem_share',
                data=df_all_states)
_ = plt.xlabel('region')
_ = plt.ylabel('percent of vote for Obama')
plt.show()
```



# Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

# Variance and standard deviation

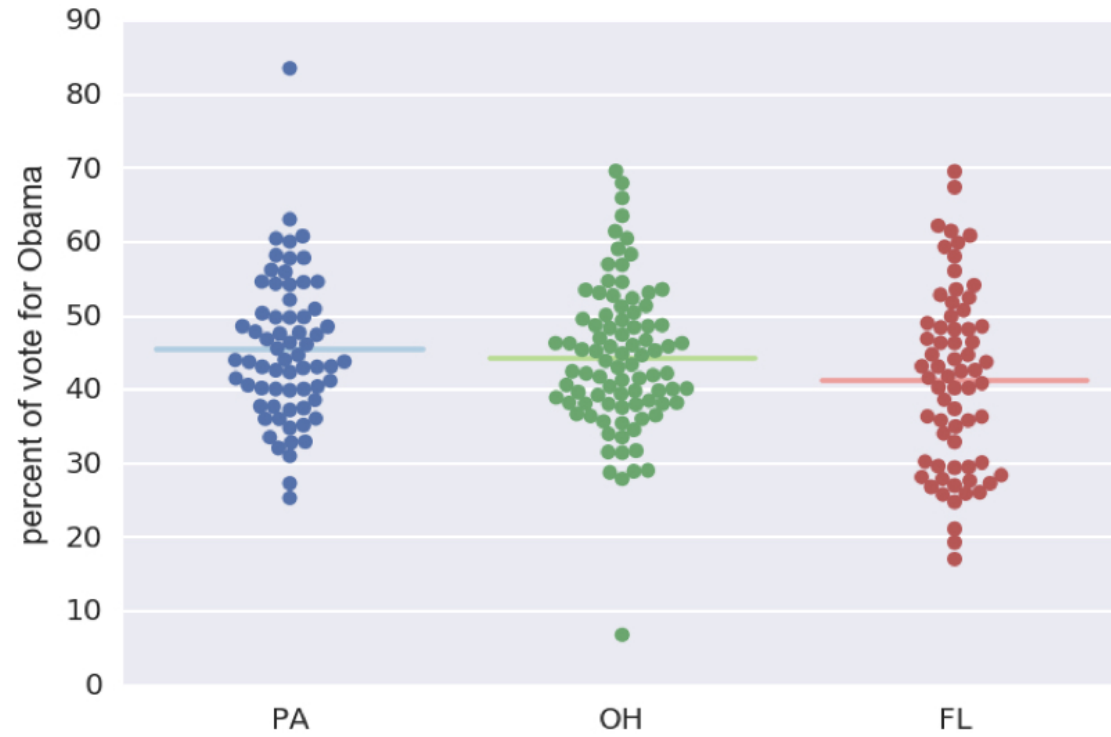
STATISTICAL THINKING IN PYTHON (PART 1)



**Justin Bois**

Lecturer at the California Institute of  
Technology

# 2008 US swing state election results

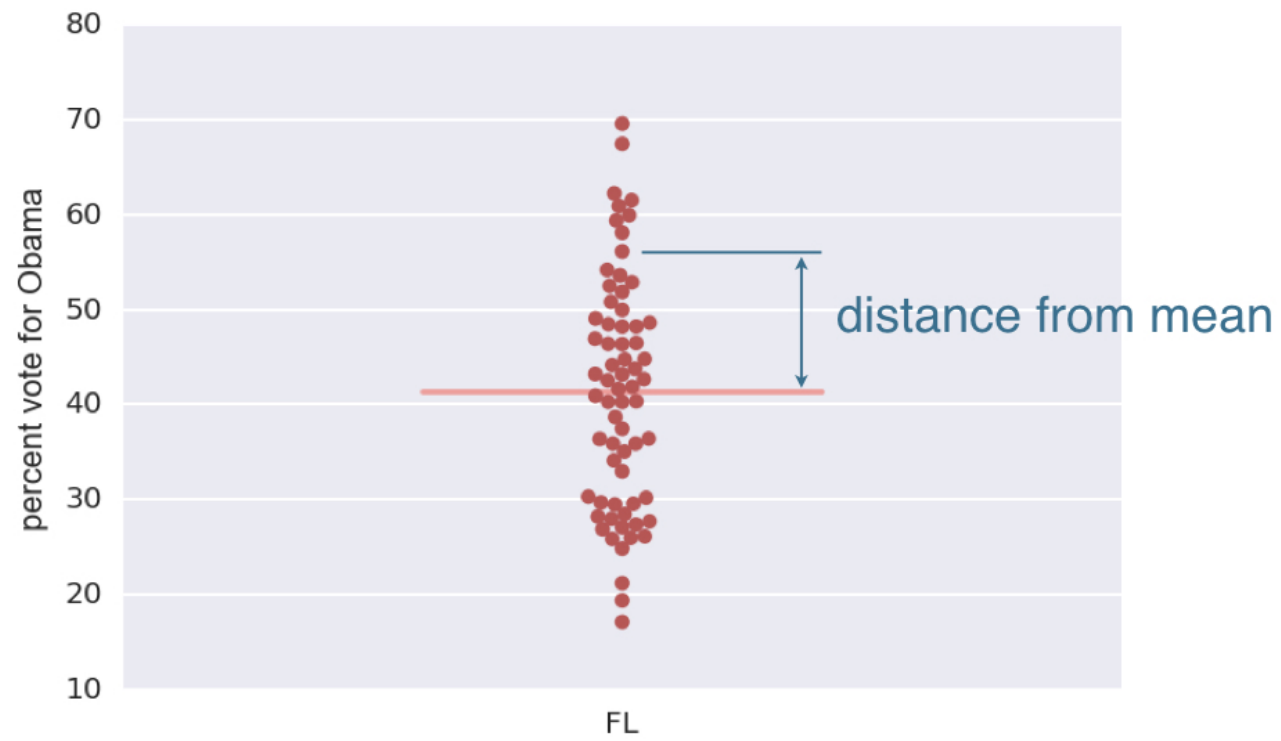


<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Variance

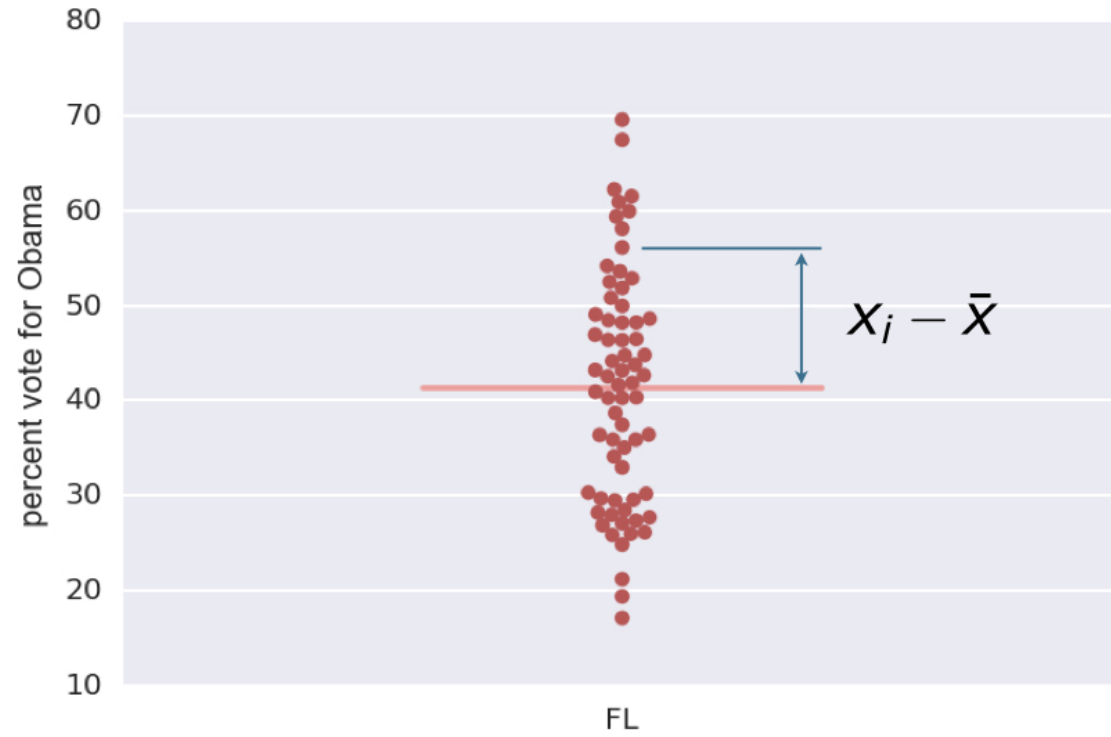
- The mean squared distance of the data from their mean
- Informally, a measure of the spread of data

# 2008 Florida election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 Florida election results



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

<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Computing the variance

```
np.var(dem_share_FL)  
147.44278618846064
```

# Computing the standard deviation

```
np.std(dem_share_FL)
```

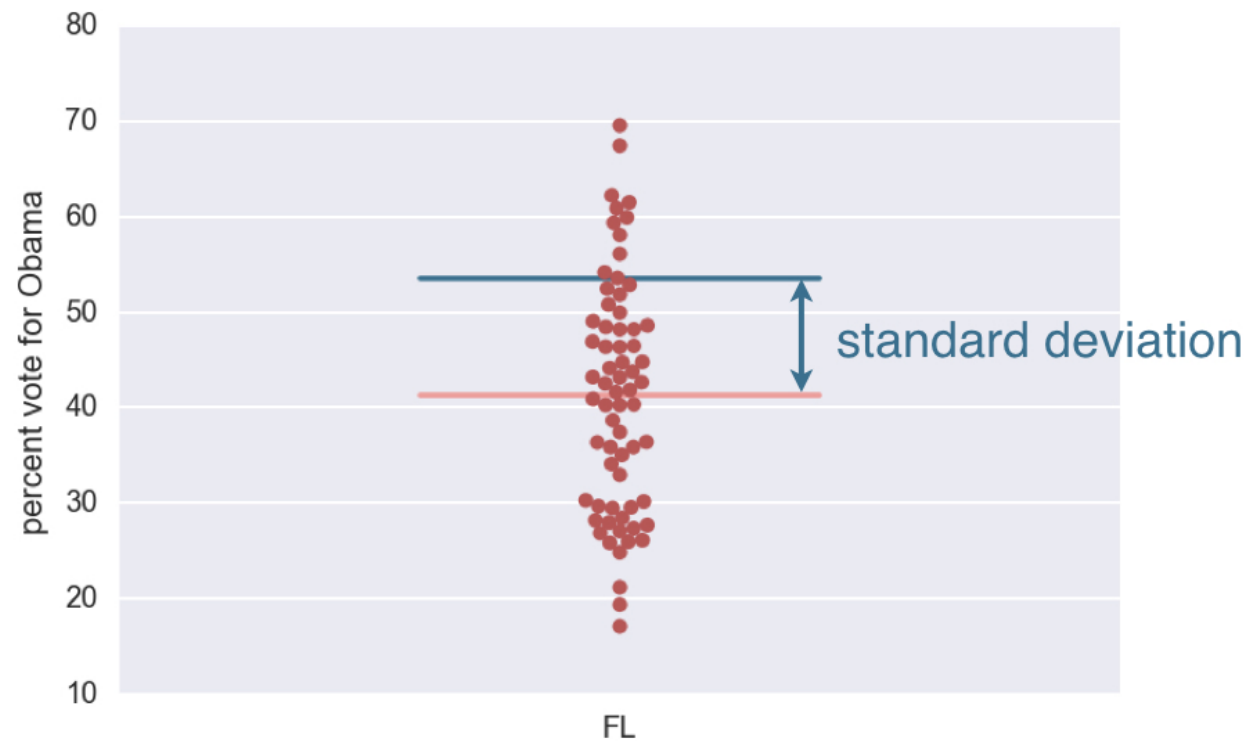
```
12.142602117687158
```

```
np.sqrt(np.var(dem_share_FL))
```

```
12.142602117687158
```



# 2008 Florida election results



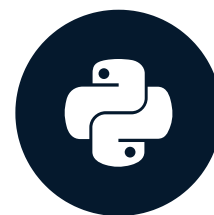
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

# Covariance and the Pearson correlation coefficient

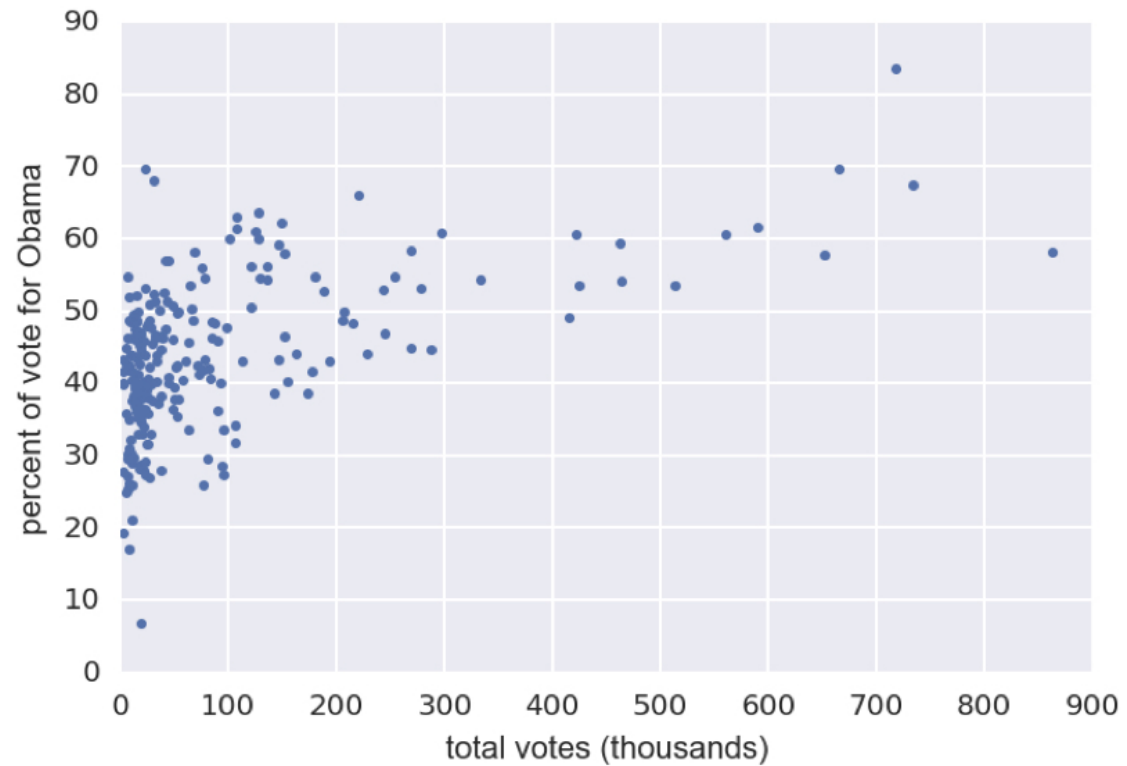
STATISTICAL THINKING IN PYTHON (PART 1)



**Justin Bois**

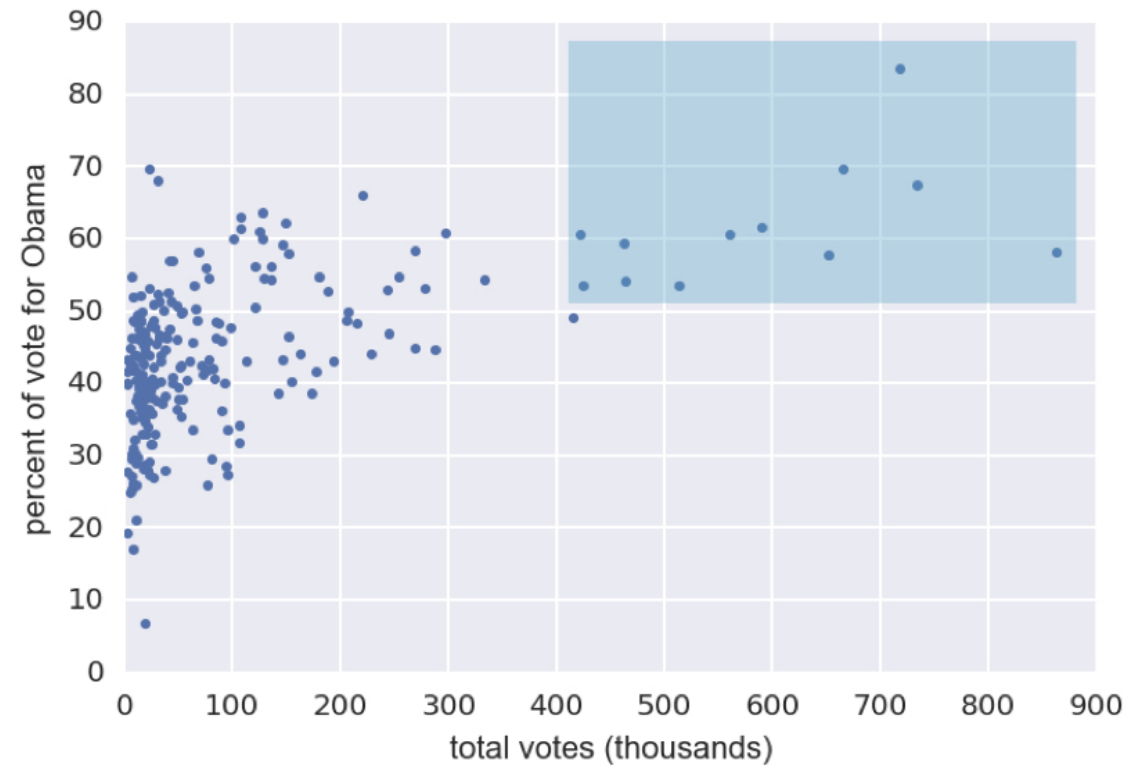
Lecturer at the California Institute of Technology

# 2008 US swing state election results



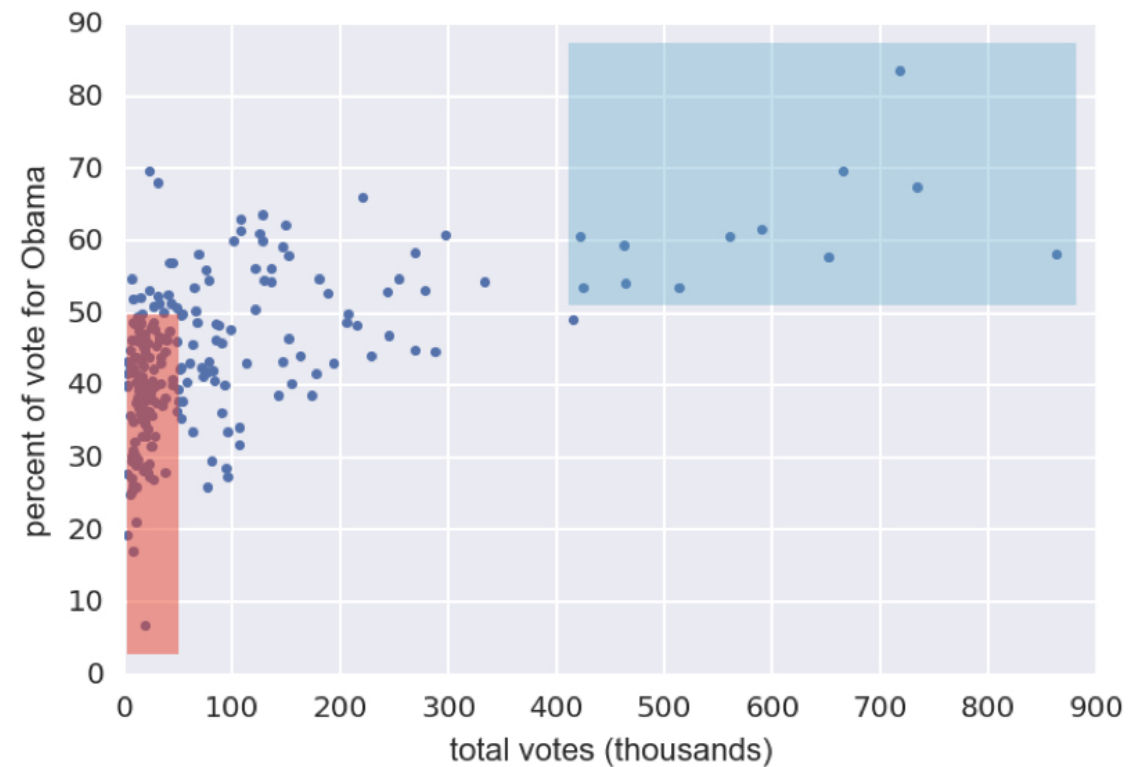
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Generating a scatter plot

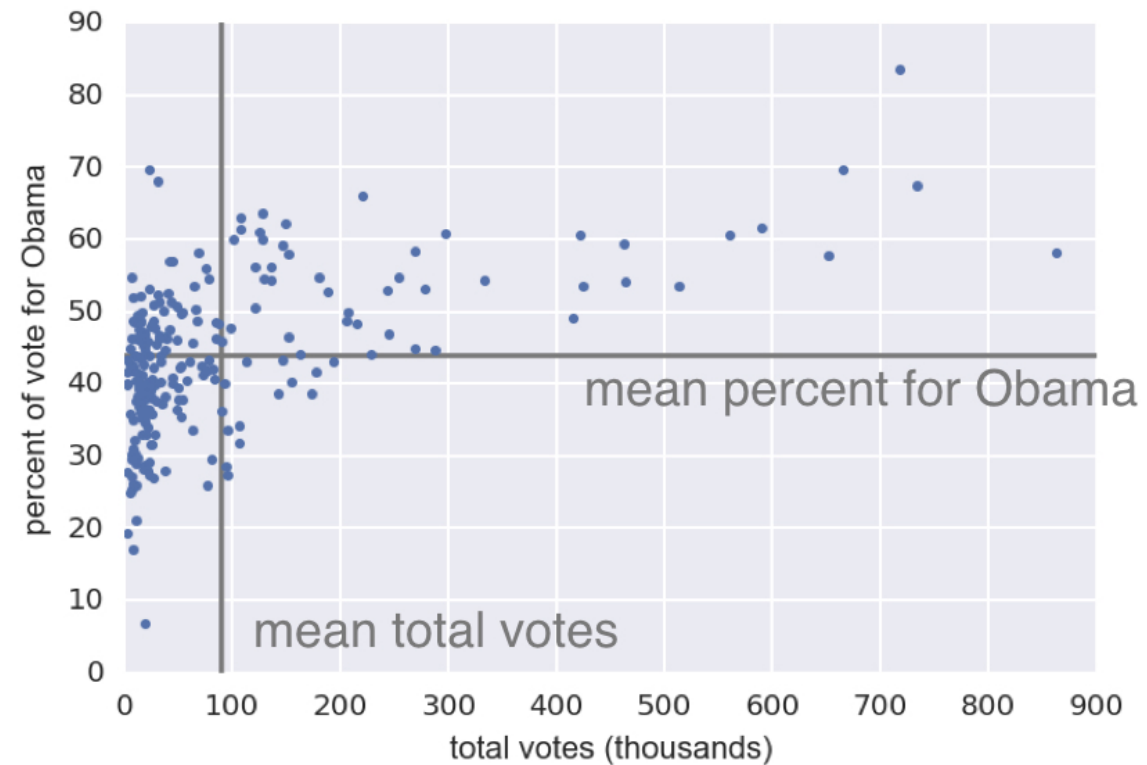
```
_ = plt.plot(total_votes/1000, dem_share,  
             marker='.', linestyle='none')  
_ = plt.xlabel('total votes (thousands)')  
_ = plt.ylabel('percent of vote for Obama')
```

# Covariance

- A measure of how two quantities vary together

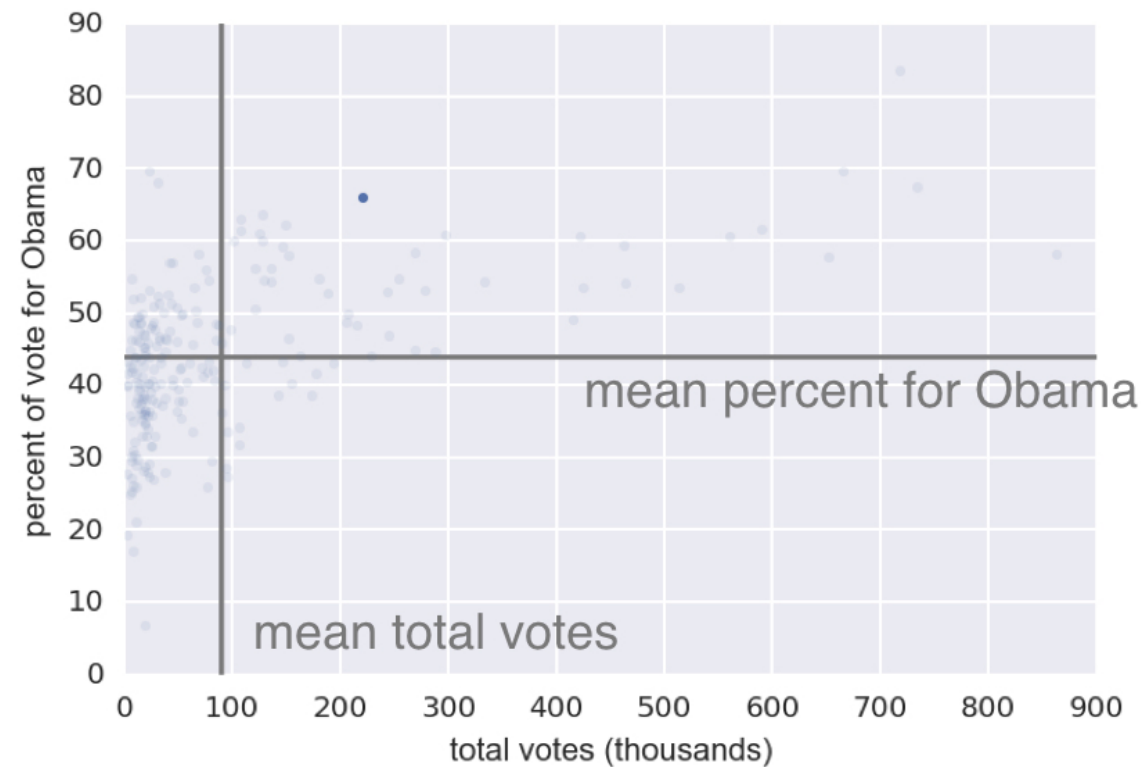


# Calculation of the covariance



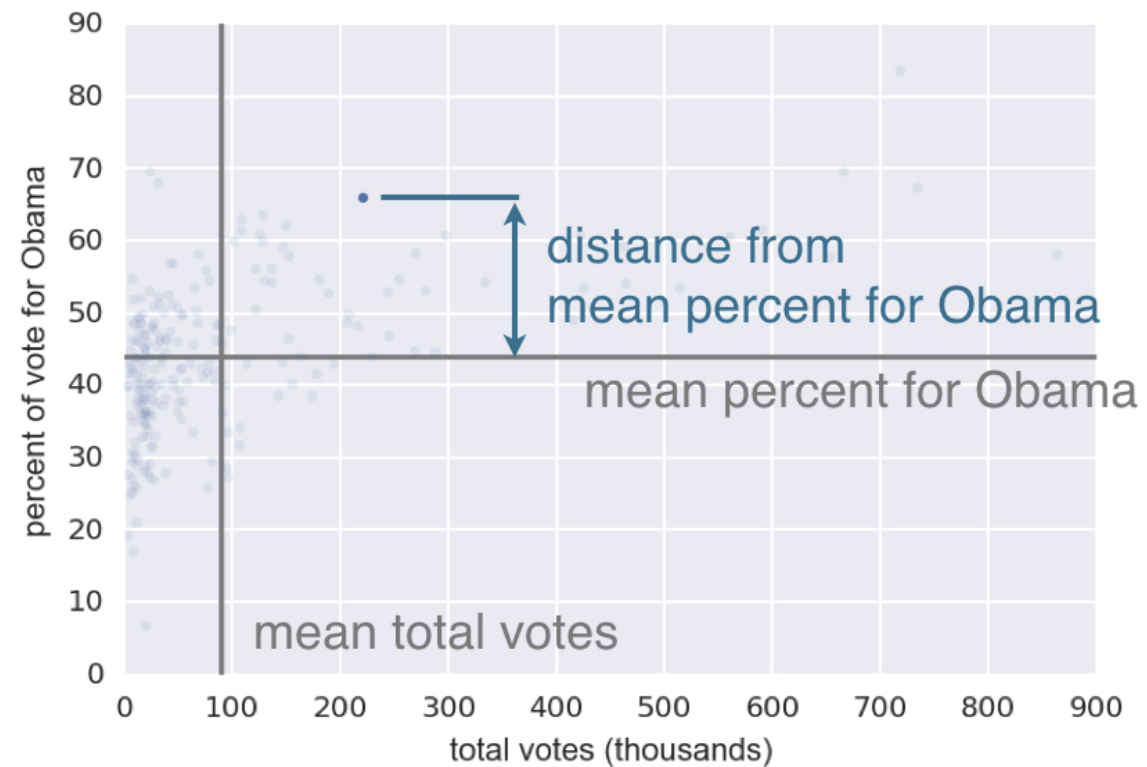
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Calculation of the covariance



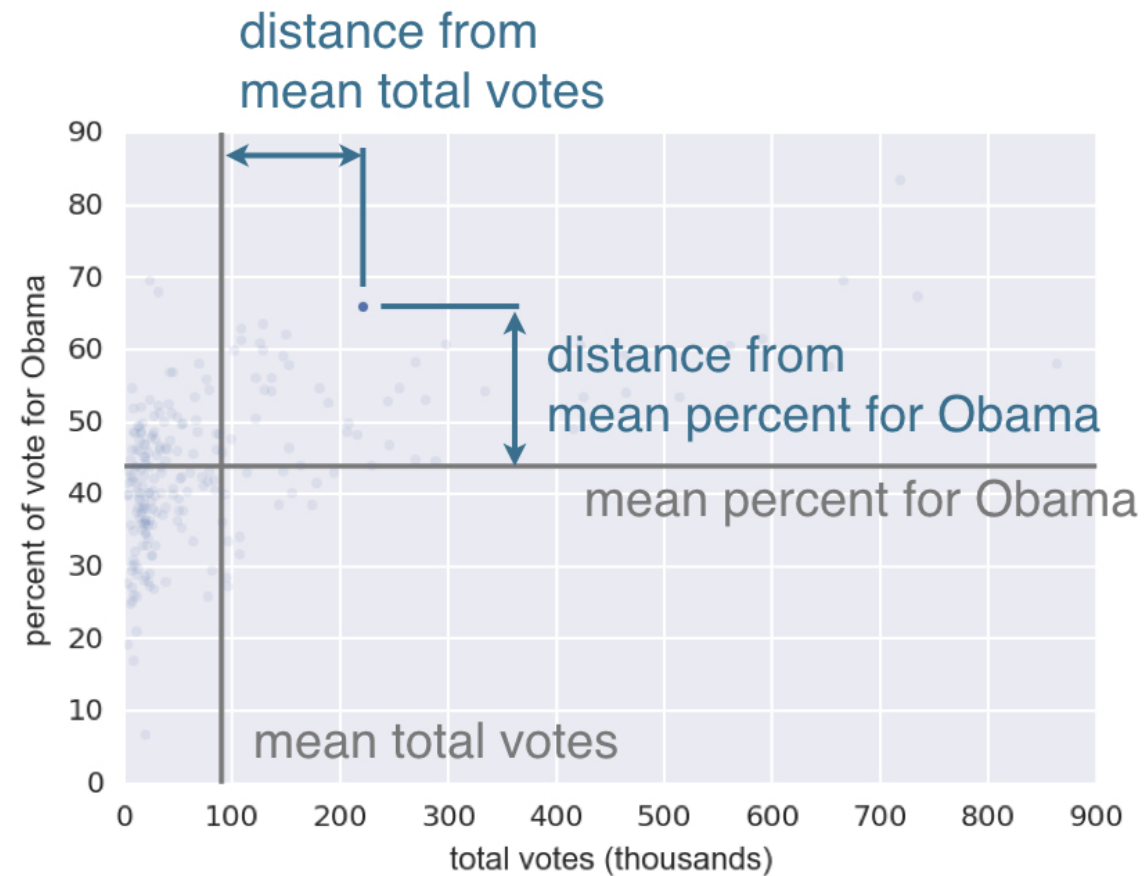
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Calculation of the covariance



<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Calculation of the covariance



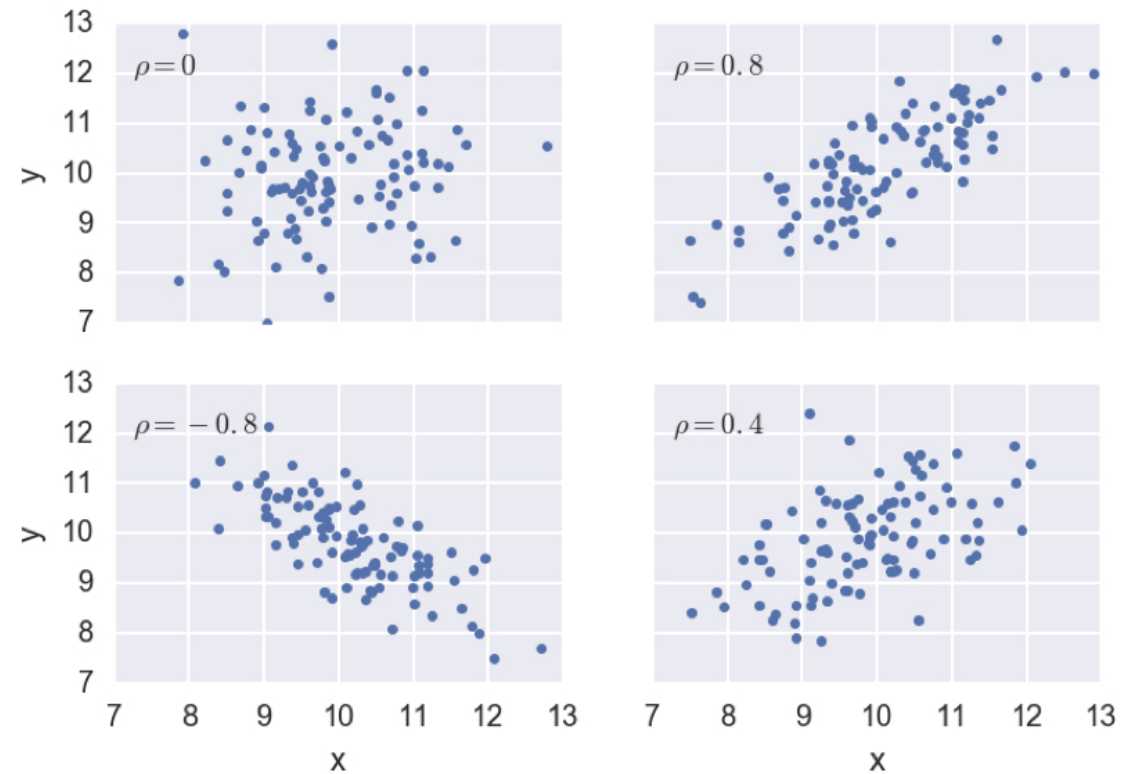
$$covariance = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# Pearson correlation coefficient

$$\rho = \text{Pearson correlation} = \frac{\text{covariance}}{(\text{std of x})(\text{std of y})}$$
$$= \frac{\text{variability due to codependence}}{\text{independant variability}}$$

# Pearson correlation coefficient examples



# Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)