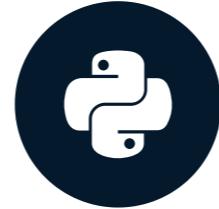


# Optimal parameters

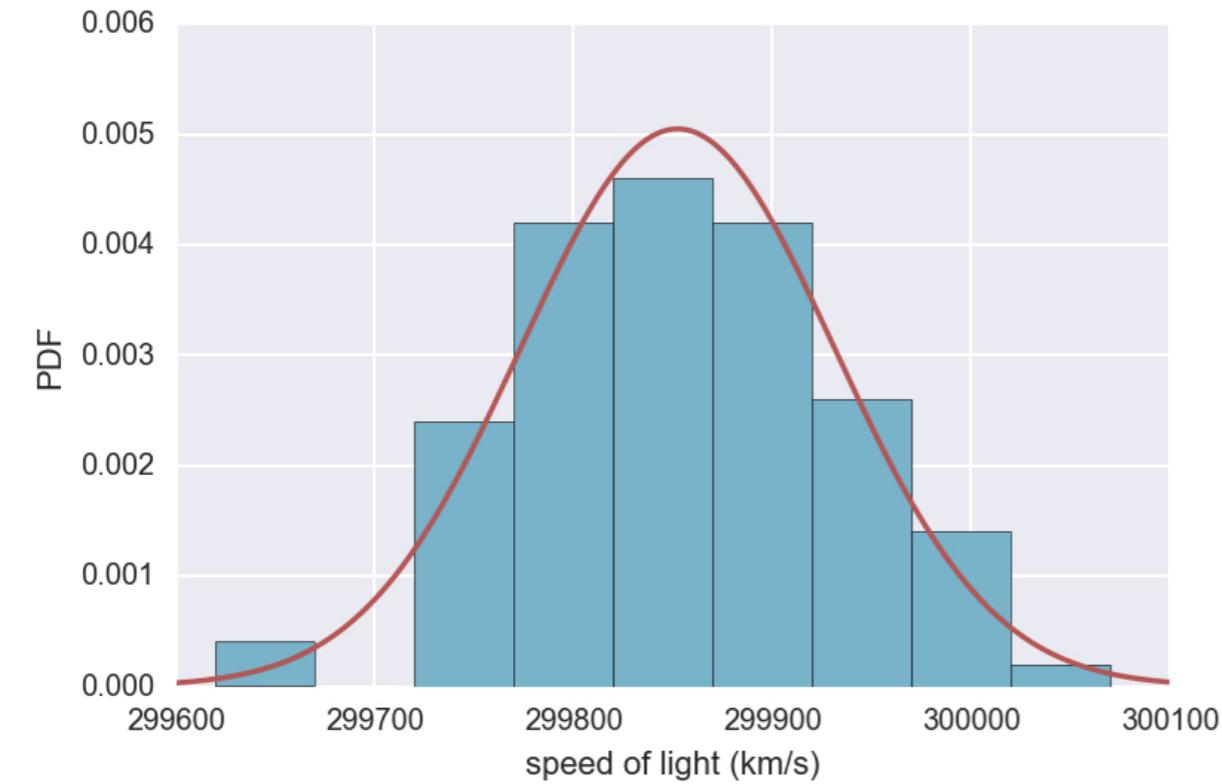
STATISTICAL THINKING IN PYTHON (PART 2)



**Justin Bois**

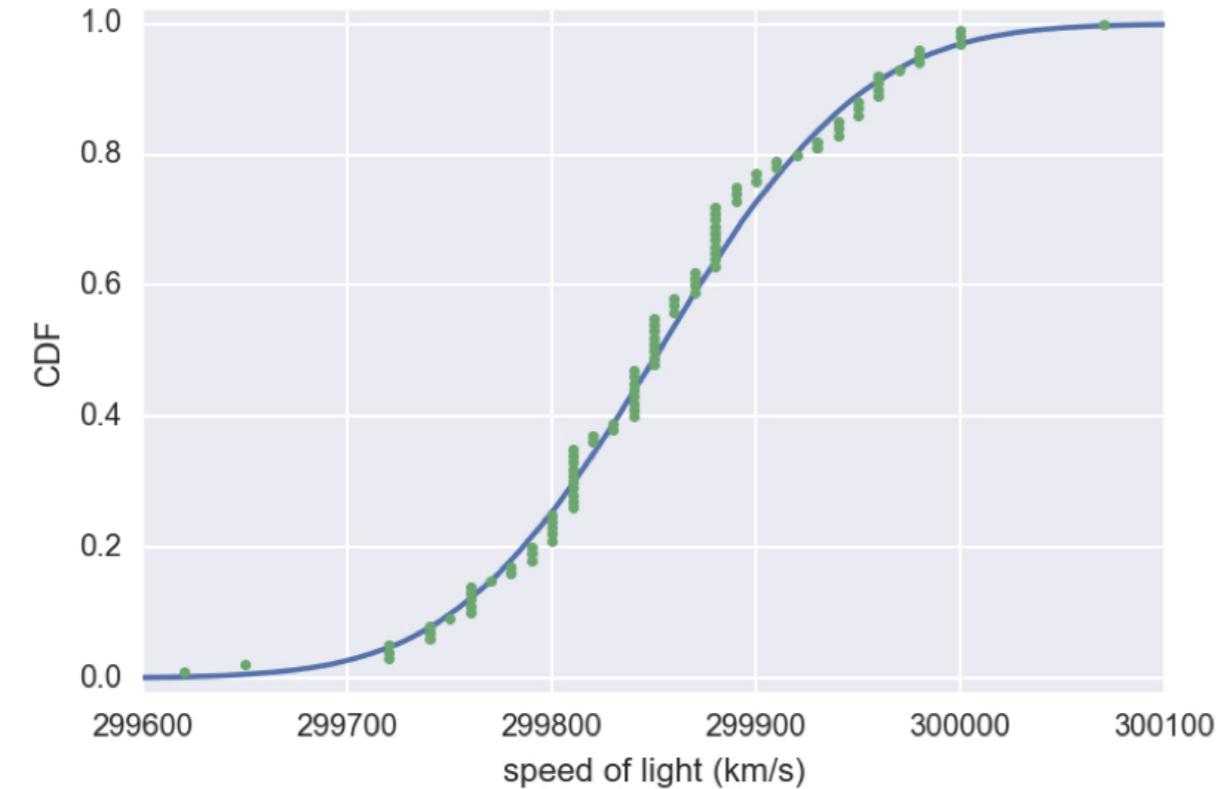
Lecturer at the California Institute of  
Technology

# Histogram of Michelson's measurements



<sup>1</sup> Data: Michelson, 1880

# CDF of Michelson's measurements

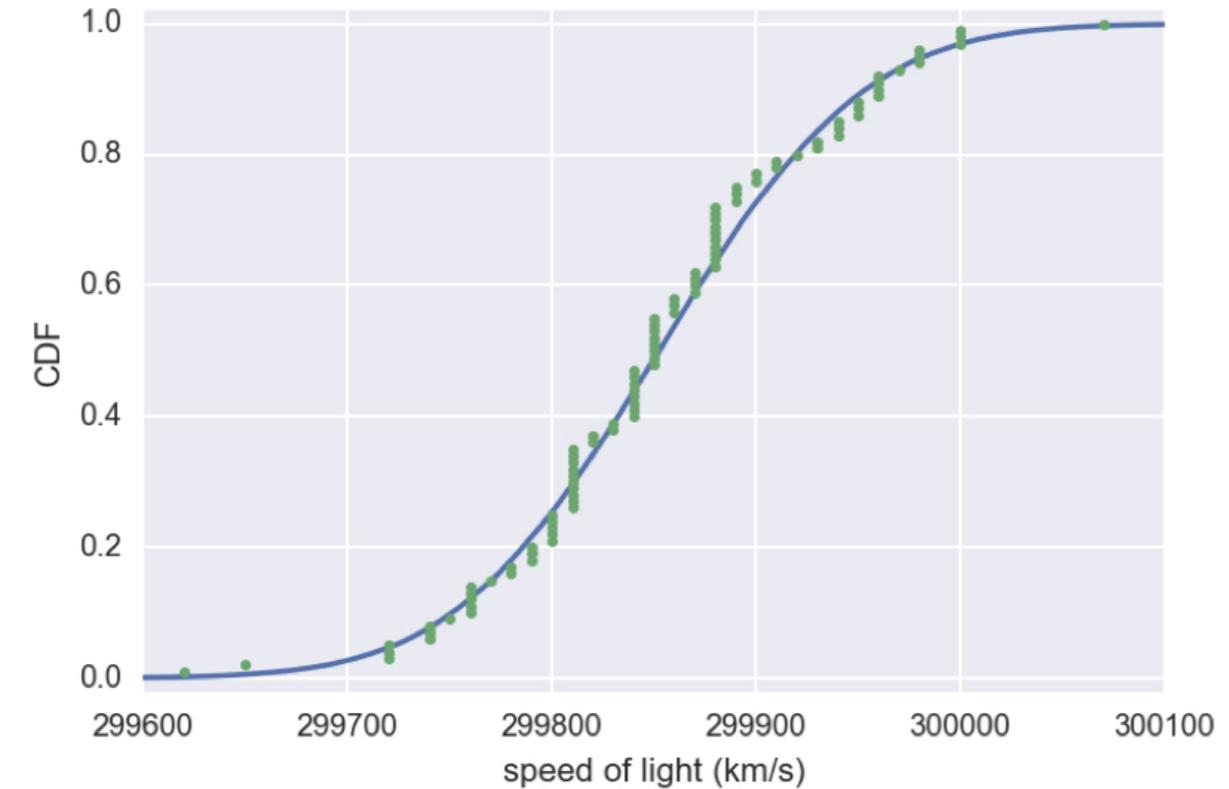


<sup>1</sup> Data: Michelson, 1880

# Checking Normality of Michelson data

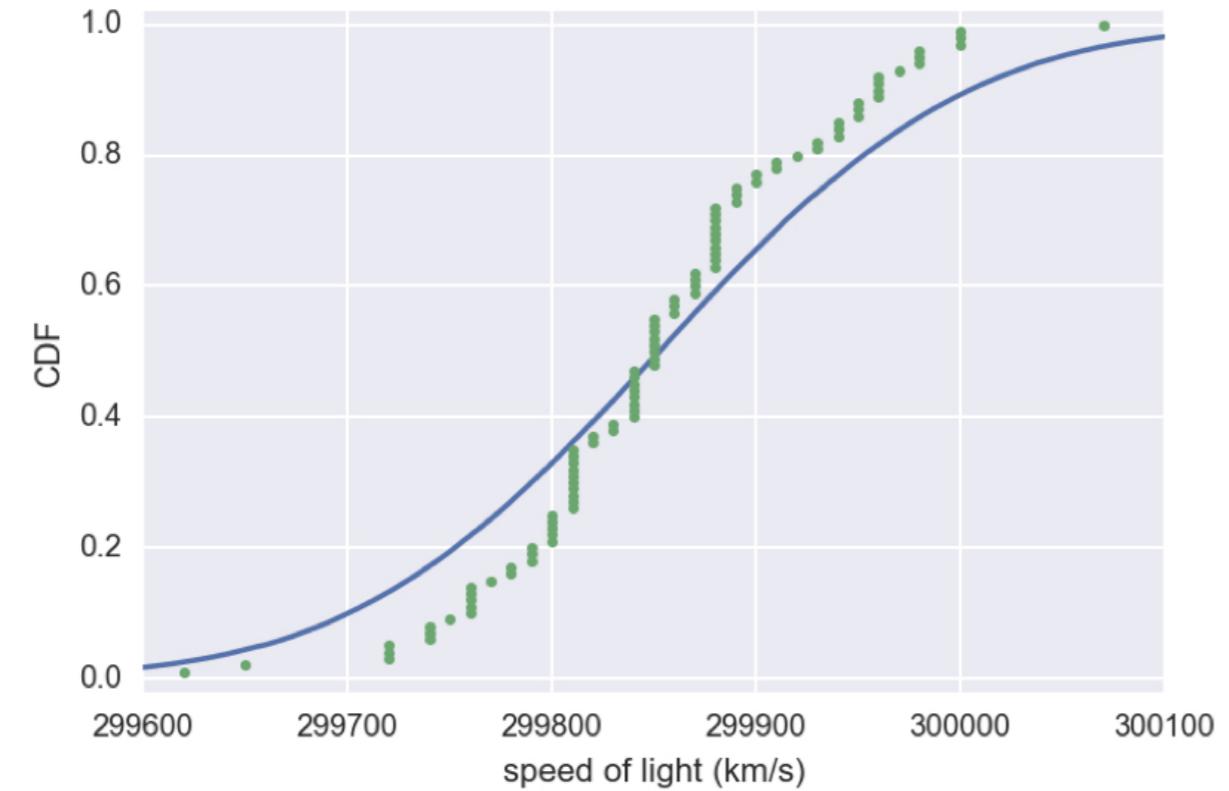
```
import numpy as np
import matplotlib.pyplot as plt
mean = np.mean(michelson_speed_of_light)
std = np.std(michelson_speed_of_light)
samples = np.random.normal(mean, std, size=10000)
```

# CDF of Michelson's measurements



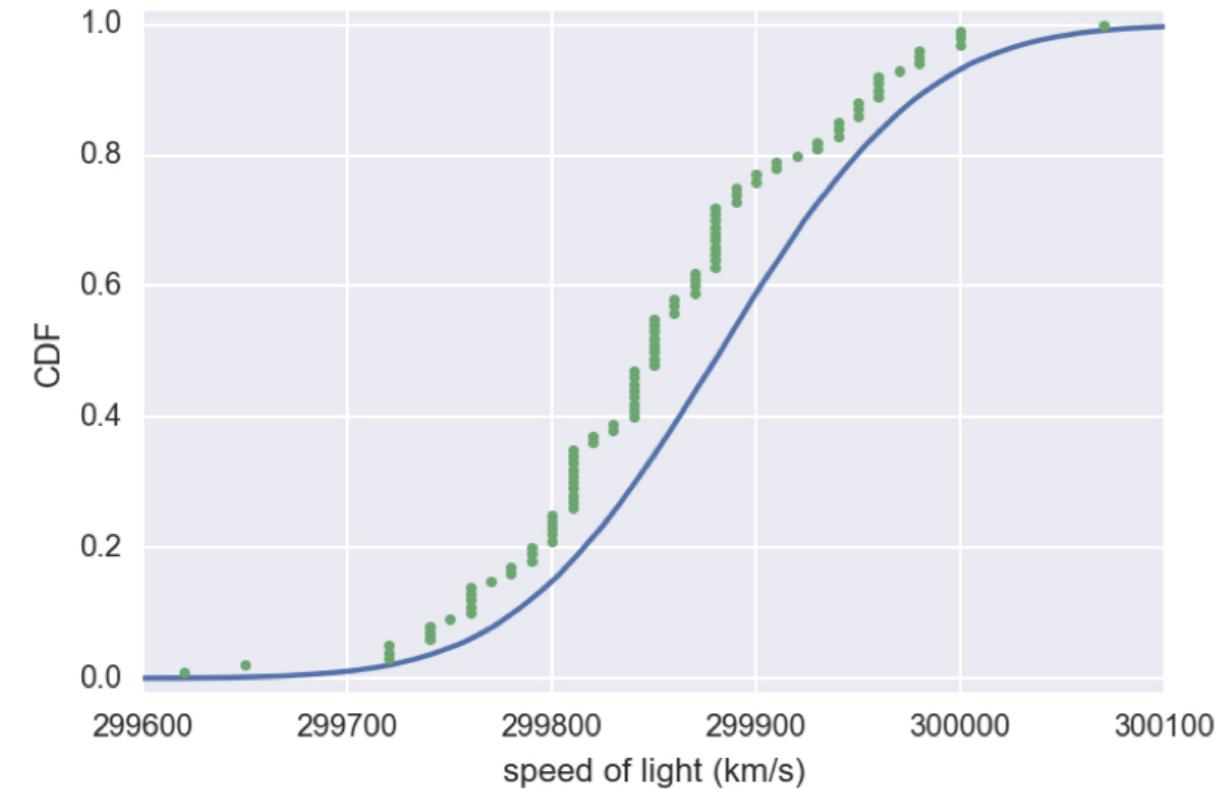
<sup>1</sup> Data: Michelson, 1880

# CDF with bad estimate of st. dev.



<sup>1</sup> Data: Michelson, 1880

# CDF with bad estimate of mean

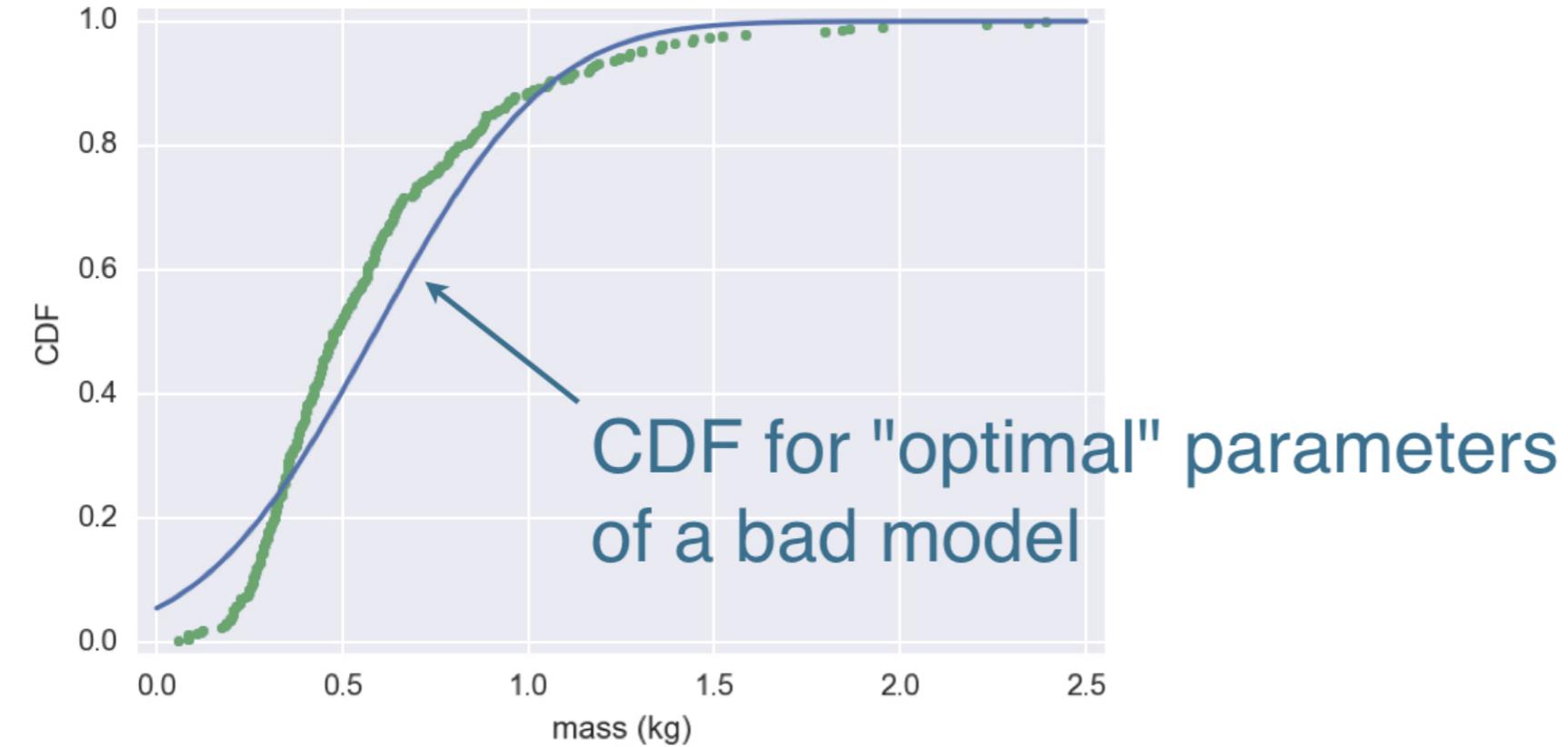


<sup>1</sup> Data: Michelson, 1880

# Optimal parameters

- Parameter values that bring the model in closest agreement with the data

# Mass of MA large mouth bass



<sup>1</sup> Source: Mass. Dept. of Environmental Protection

# Packages to do statistical inference



scipy.stats

# Packages to do statistical inference



scipy.stats



statsmodels

# Packages to do statistical inference



scipy.stats



statsmodels



hacker stats  
with numpy

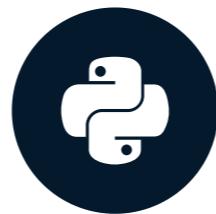
<sup>1</sup> Knife image: D-M Commons, CC BY-SA 3.0

# **Let's practice!**

**STATISTICAL THINKING IN PYTHON (PART 2)**

# Linear regression by least squares

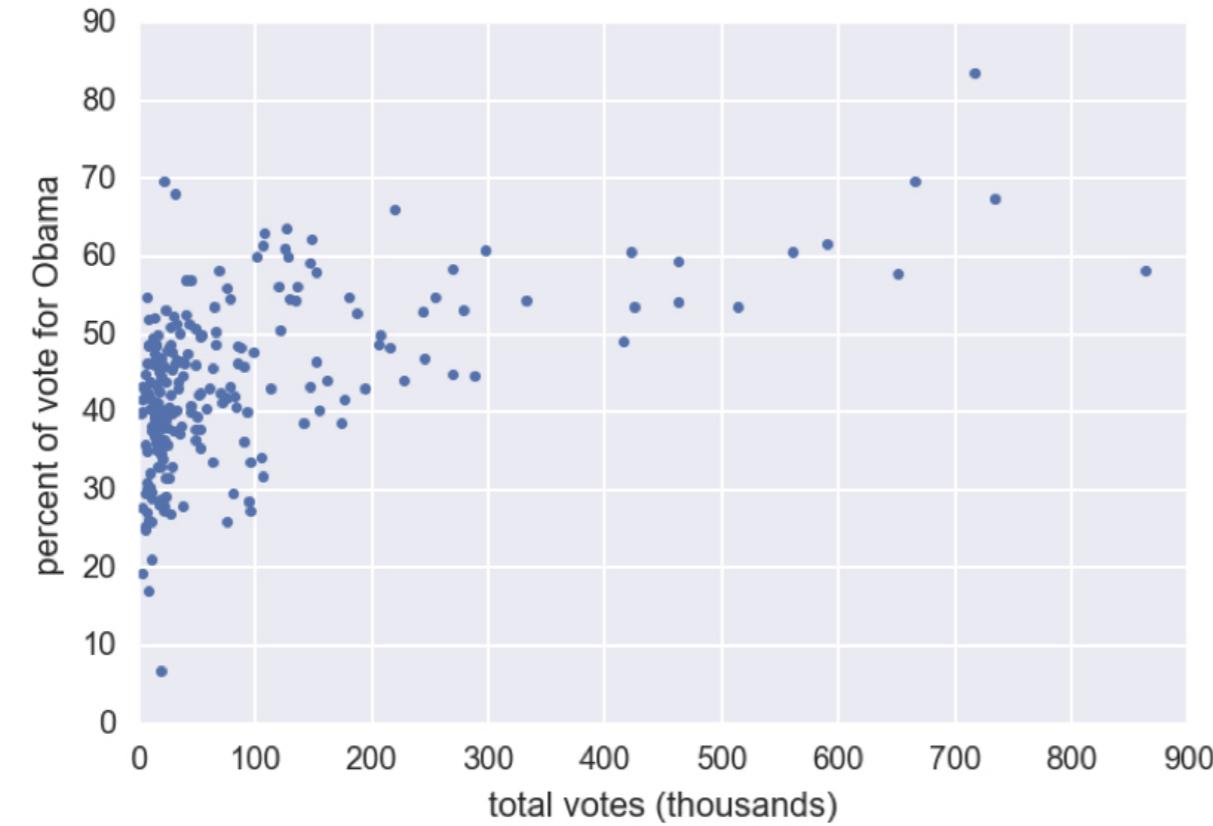
STATISTICAL THINKING IN PYTHON (PART 2)



Justin Bois

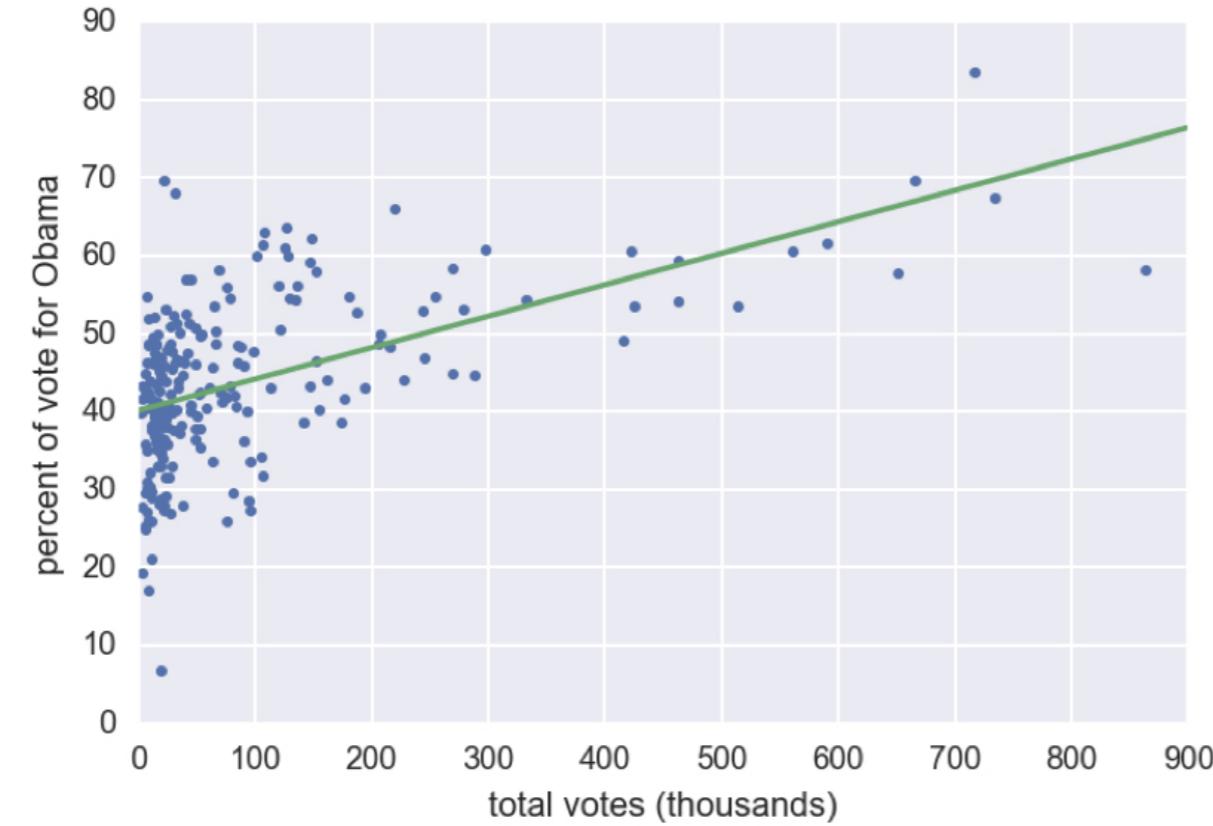
Lecturer at the California Institute of  
Technology

# 2008 US swing state election results



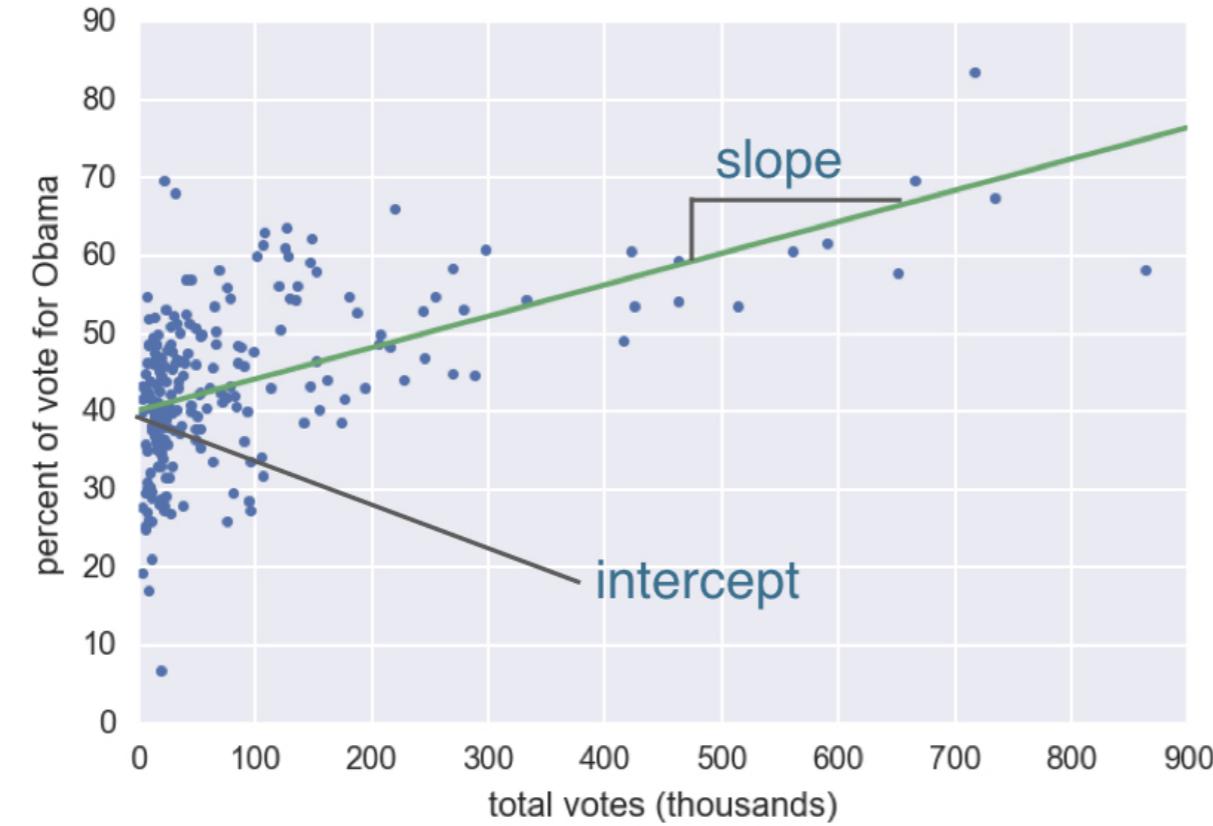
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



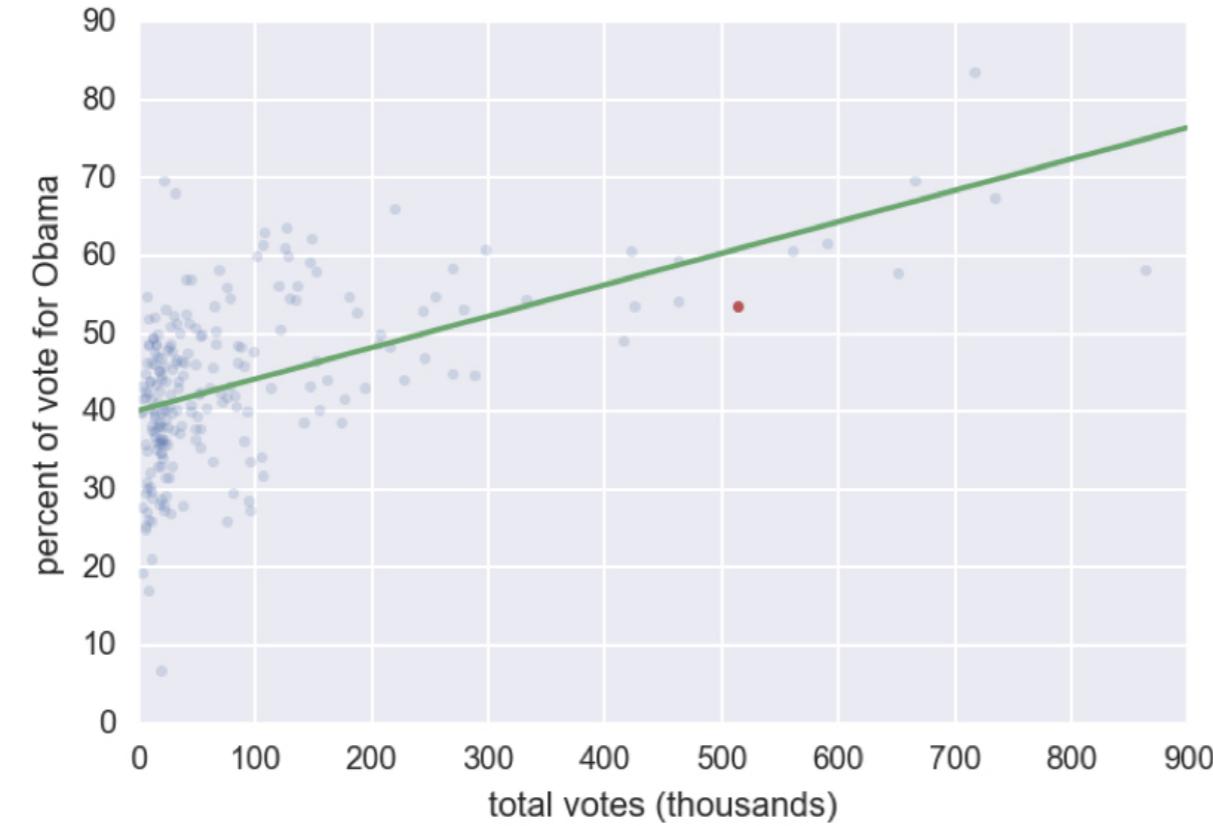
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



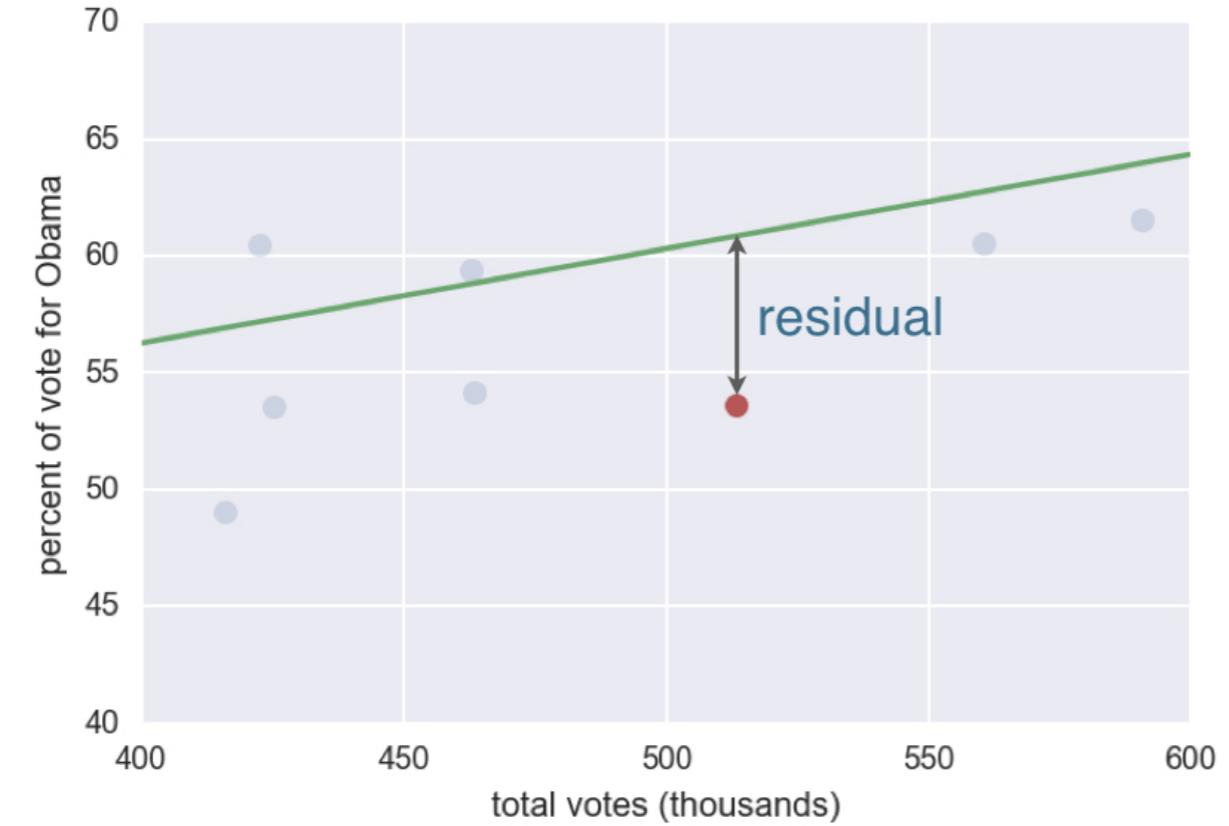
<sup>1</sup> Data retrieved from Data.gov (<https://www.data.gov/>)

# 2008 US swing state election results



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

# Residuals



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

# Least squares

- The process of finding the parameters for which the sum of the squares of the residuals is minimal

# Least squares with np.polyfit()

```
slope, intercept = np.polyfit(total_votes,  
                               dem_share, 1)
```

slope

```
4.037071700946555e-05
```

intercept

```
40.113911968641744
```

# **Let's practice!**

**STATISTICAL THINKING IN PYTHON (PART 2)**

# The importance of EDA: Anscombe's quartet

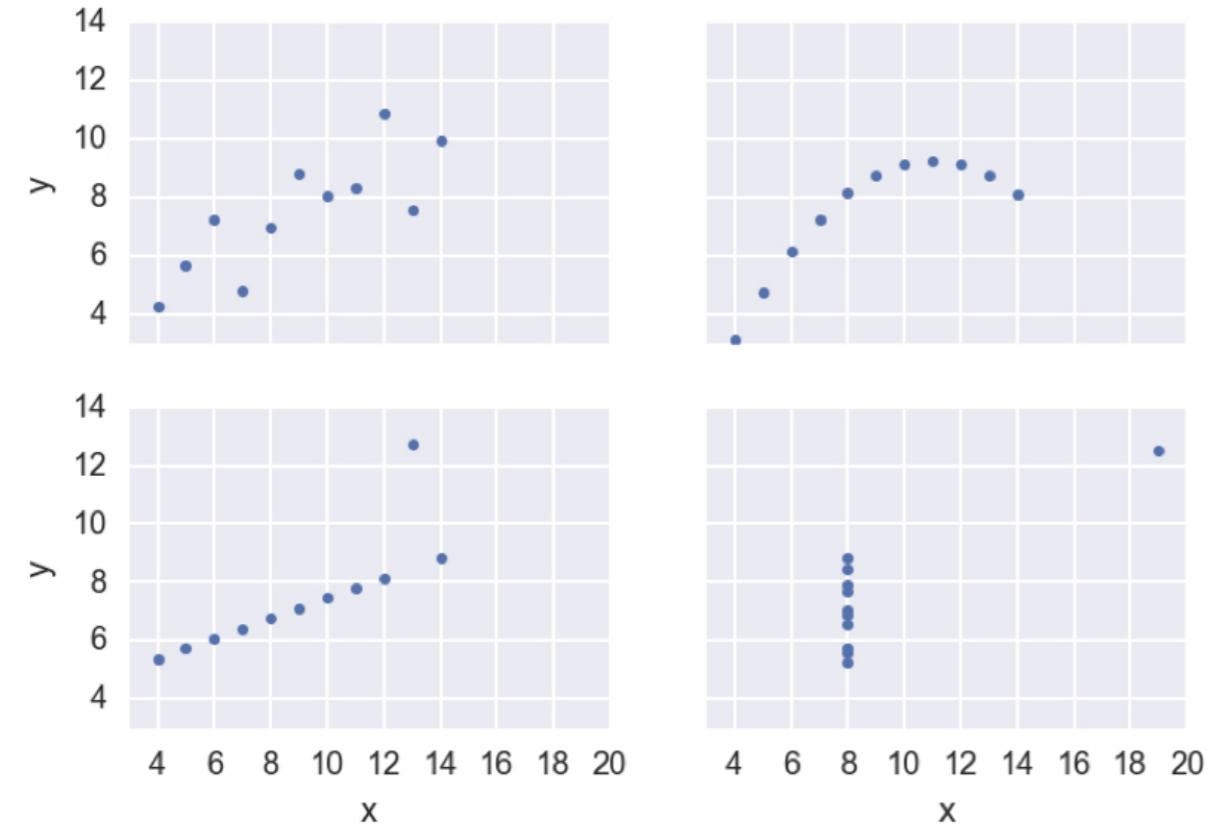
STATISTICAL THINKING IN PYTHON (PART 2)

Justin Bois

Lecturer at the California Institute of  
Technology

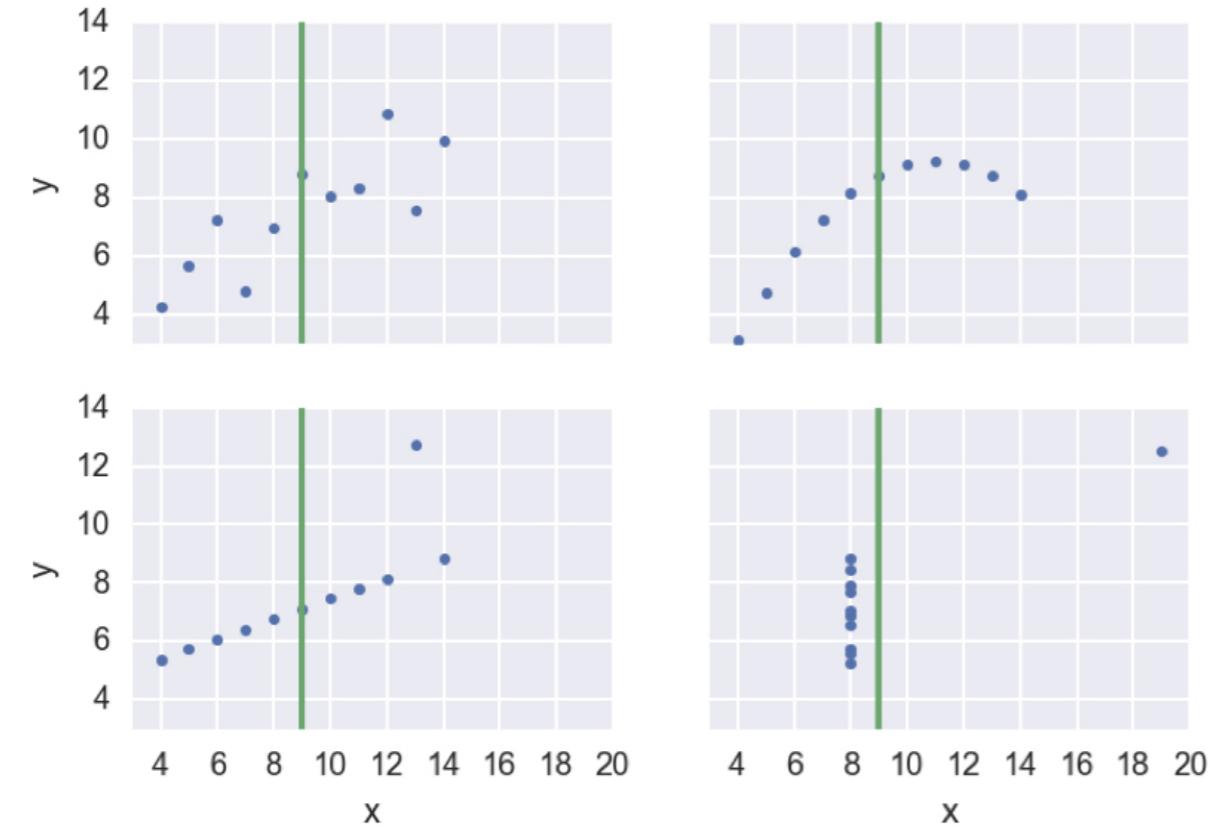


# Anscombe's quartet



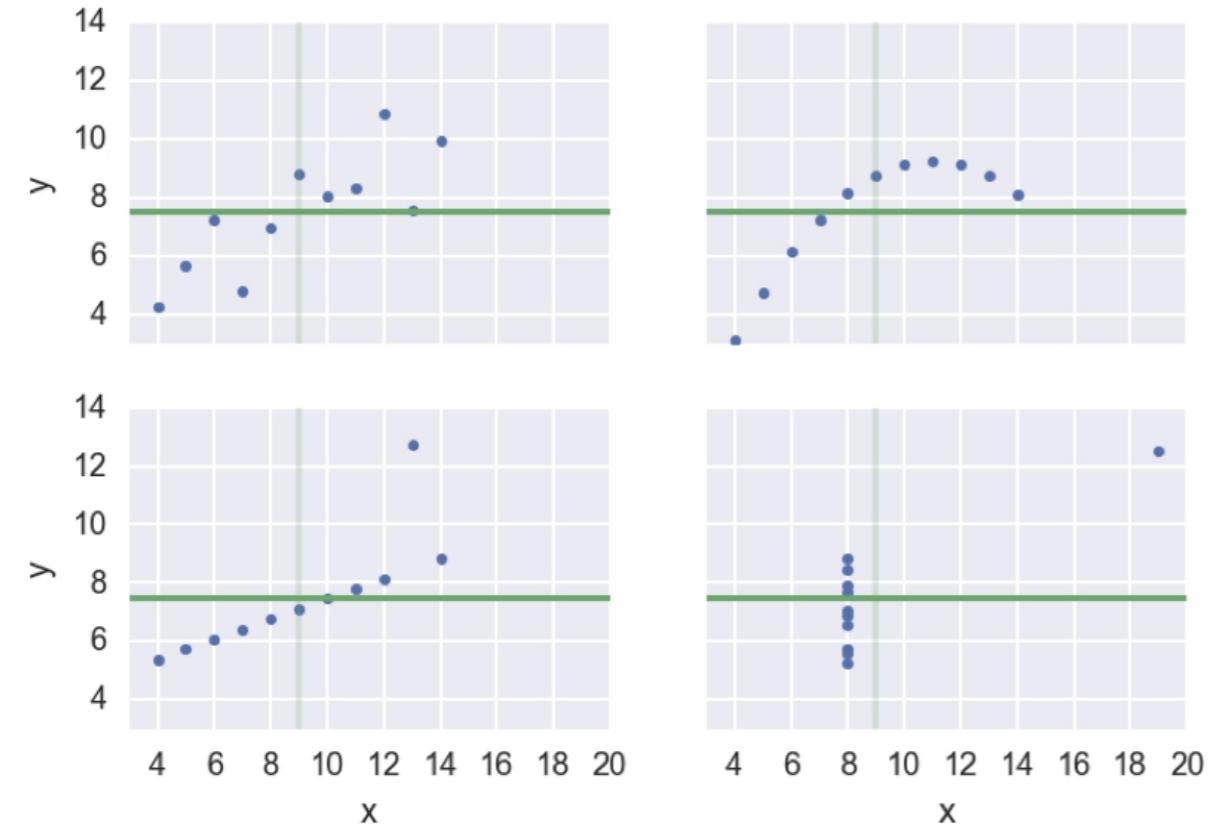
<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



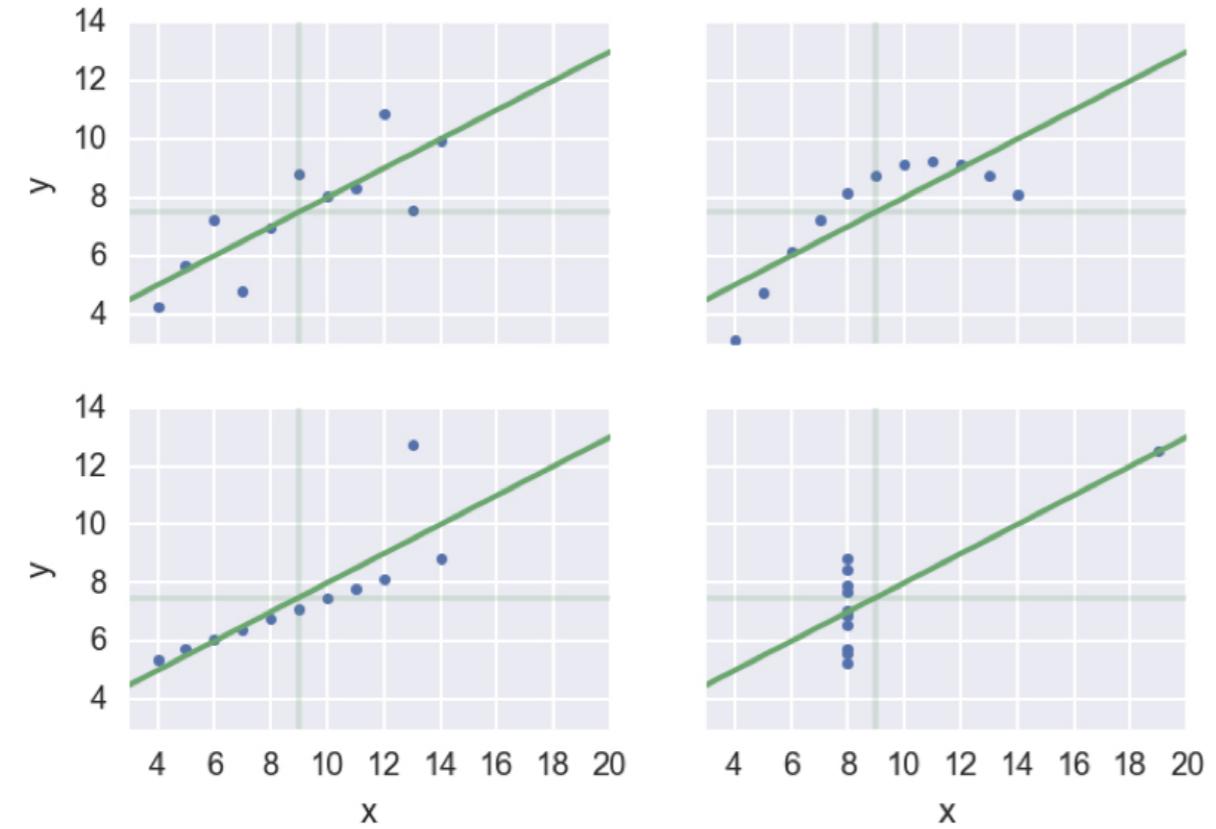
<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



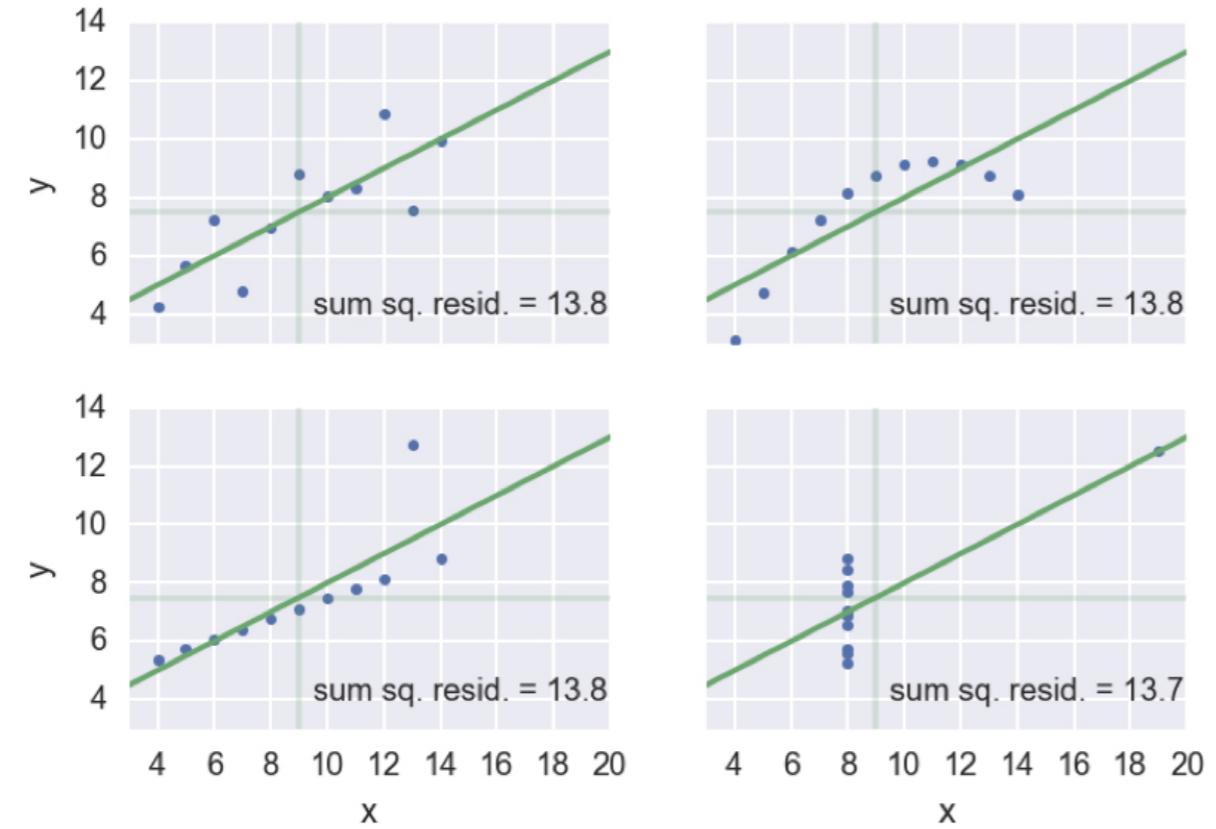
<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet

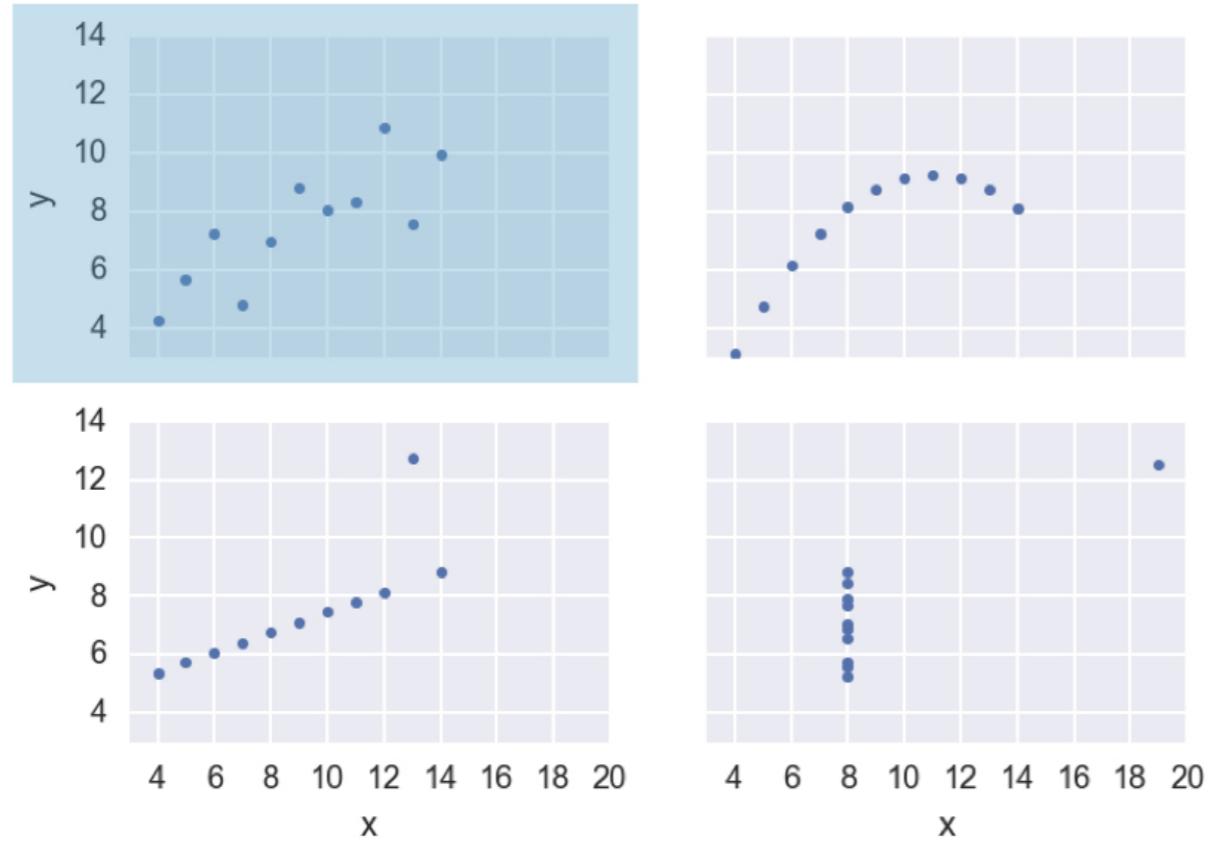


<sup>1</sup> Data: Anscombe, The American Statistician, 1973

# Look before you leap!

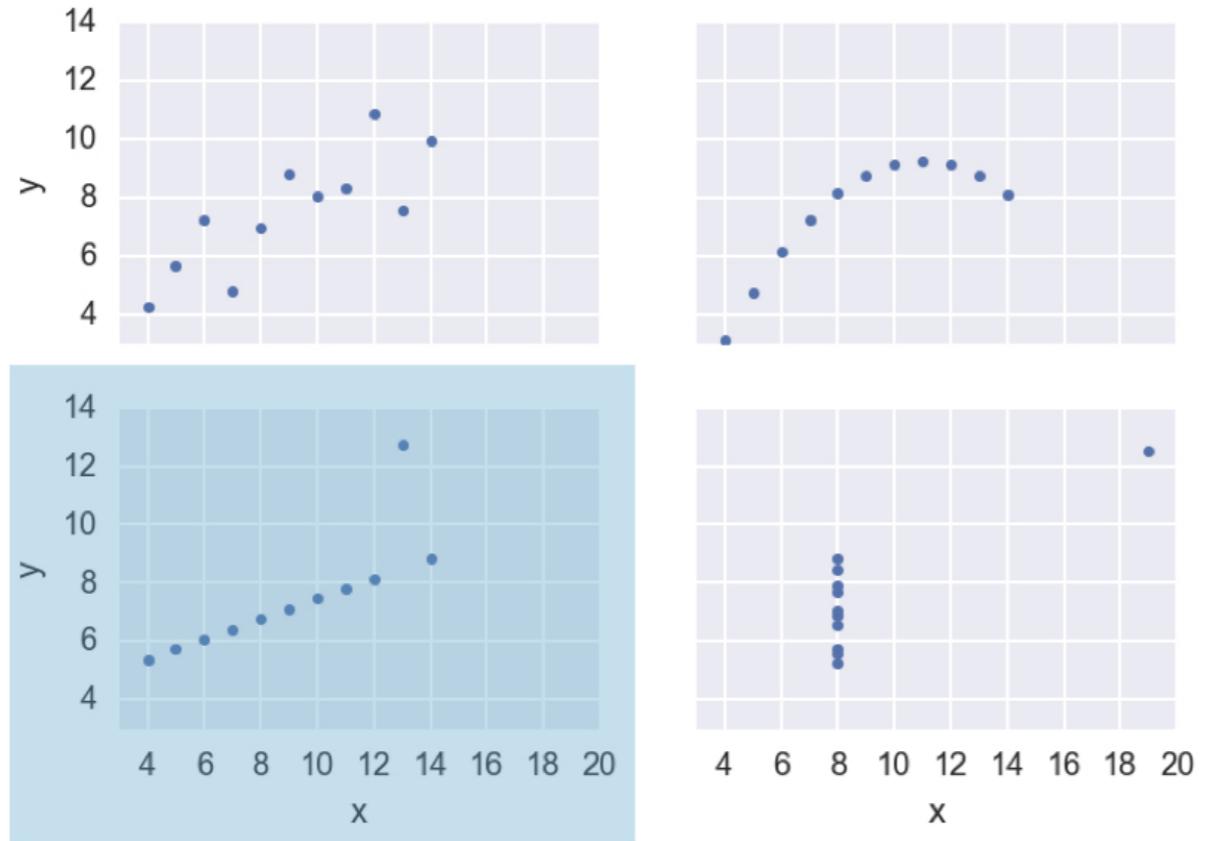
- Do graphical EDA first

# Anscombe's quartet



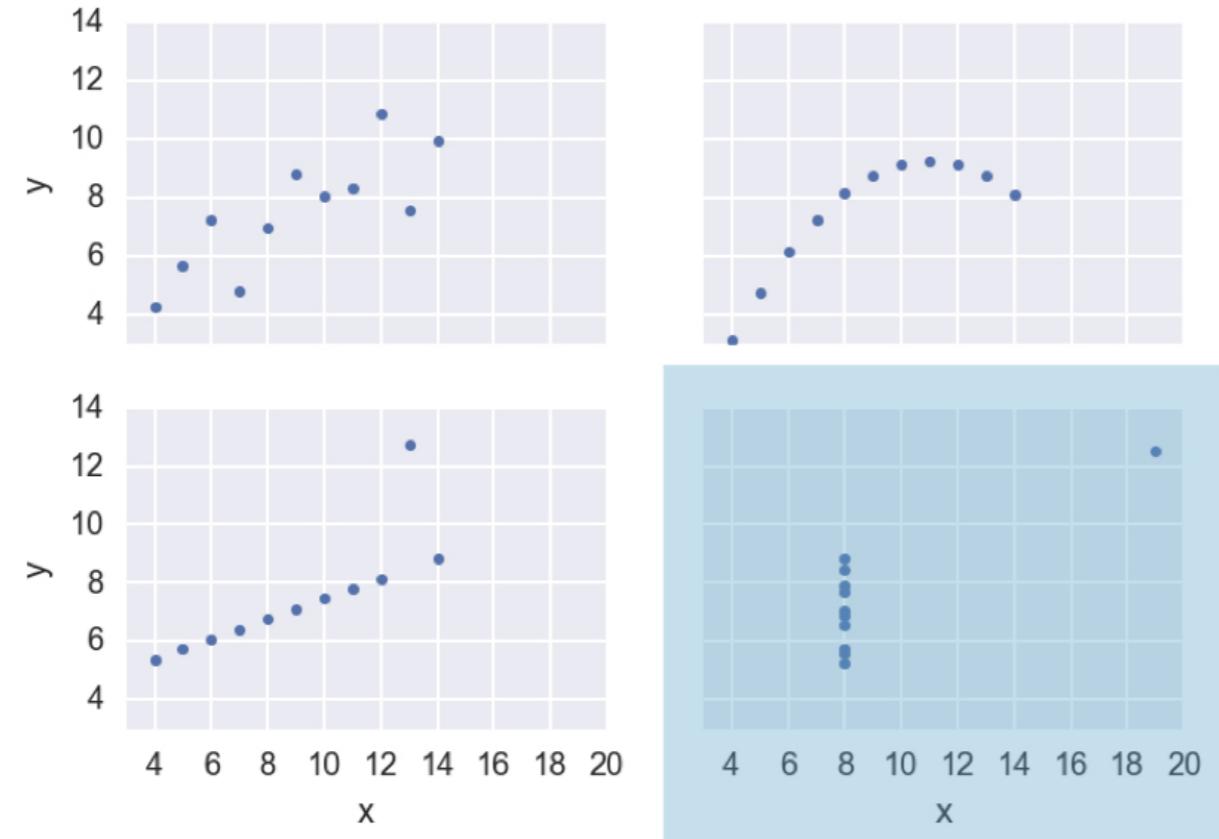
<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



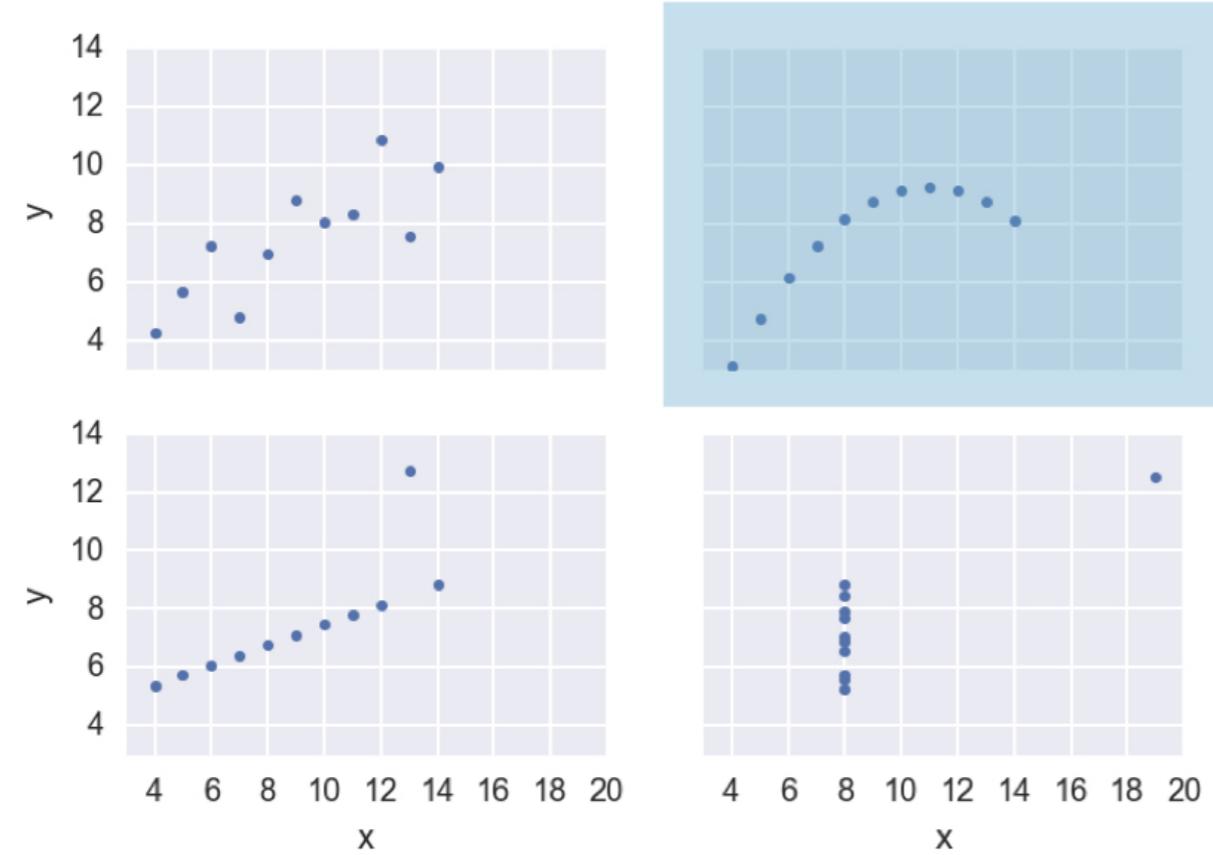
<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# Anscombe's quartet



<sup>1</sup> Data: Anscombe, *The American Statistician*, 1973

# **Let's practice!**

**STATISTICAL THINKING IN PYTHON (PART 2)**