Probability density functions

STATISTICAL THINKING IN PYTHON (PART 1)

Justin Bois

Lecturer at the California Institute of Technology

Continuous variables

- Quantities that can take any value, not just discrete values

Michelson's speed of light experiment

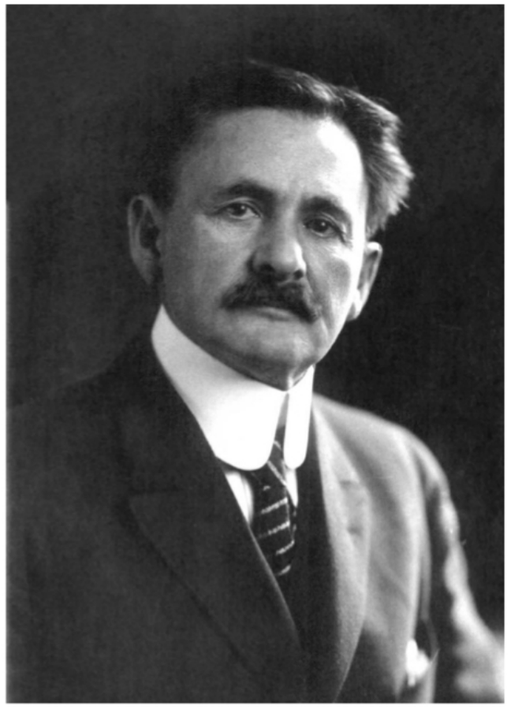


Image: public domain, Smithsonian

Data: Michelson, 1880

Michelson's speed of light experiment

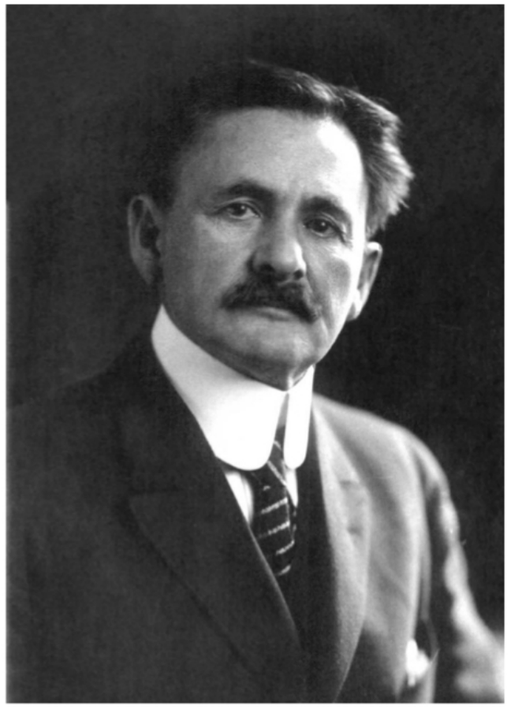


Image: public domain, Smithsonian

Data: Michelson, 1880

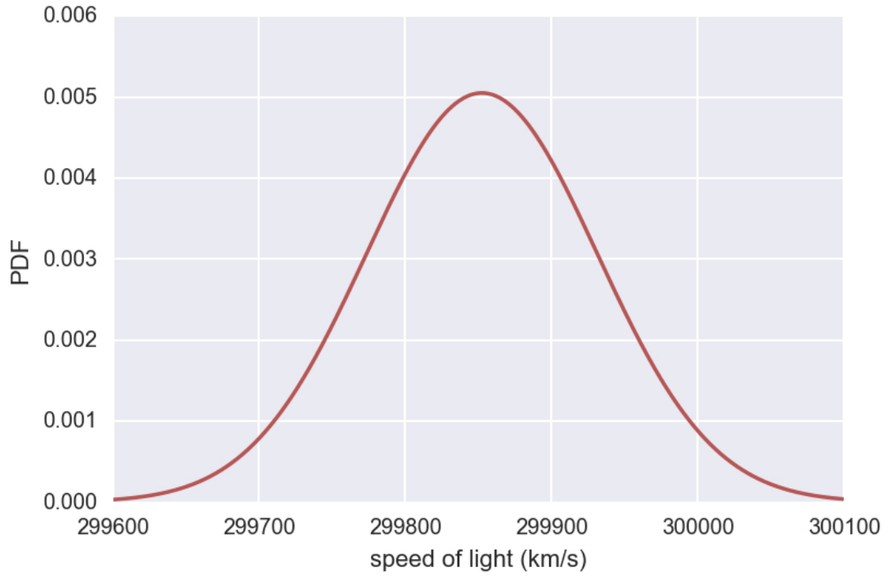
measured speed of light (1000 km/s)

299.85	299.74	299.90	300.07	299.93
299.85	299.95	299.98	299.98	299.88
300.00	299.98	299.93	299.65	299.76
299.81	300.00	300.00	299.96	299.96
299.96	299.94	299.96	299.94	299.88
299.80	299.85	299.88	299.90	299.84
299.83	299.79	299.81	299.88	299.88
299.83	299.80	299.79	299.76	299.80
299.88	299.88	299.88	299.86	299.72
299.72	299.62	299.86	299.97	299.95
299.88	299.91	299.85	299.87	299.84
299.84	299.85	299.84	299.84	299.84
299.89	299.81	299.81	299.82	299.80
299.77	299.76	299.74	299.75	299.76
299.91	299.92	299.89	299.86	299.88
299.72	299.84	299.85	299.85	299.78
299.89	299.84	299.78	299.81	299.76
299.81	299.79	299.81	299.82	299.85
299.87	299.87	299.81	299.74	299.81
299.94	299.95	299.80	299.81	299.87

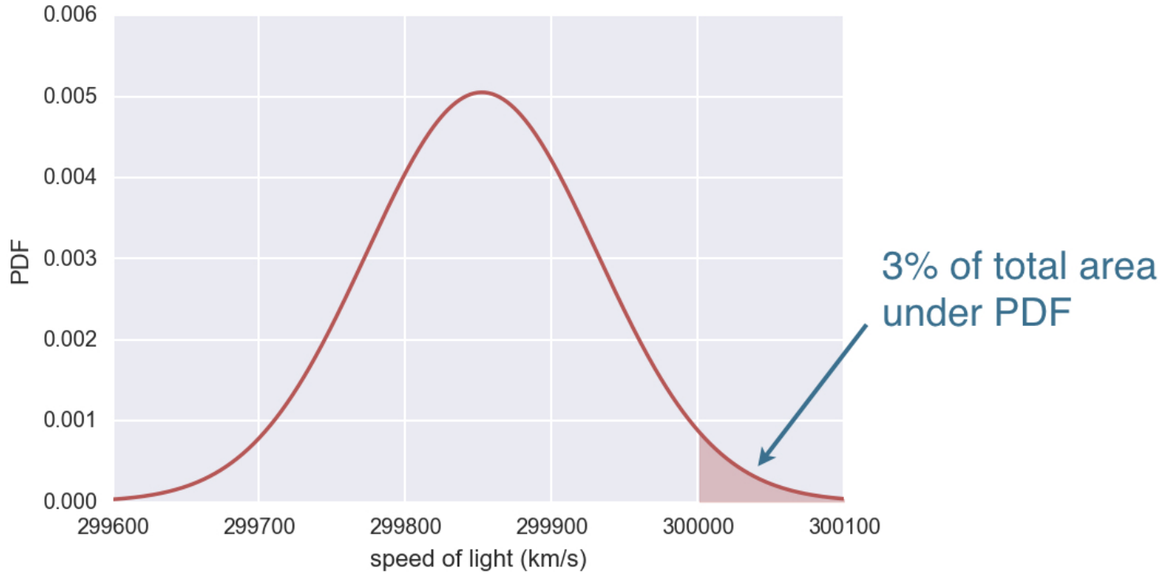
Probability density function (PDF)

- Continuous analog to the PMF
- Mathematical description of the relative likelihood of observing a value of a continuous variable

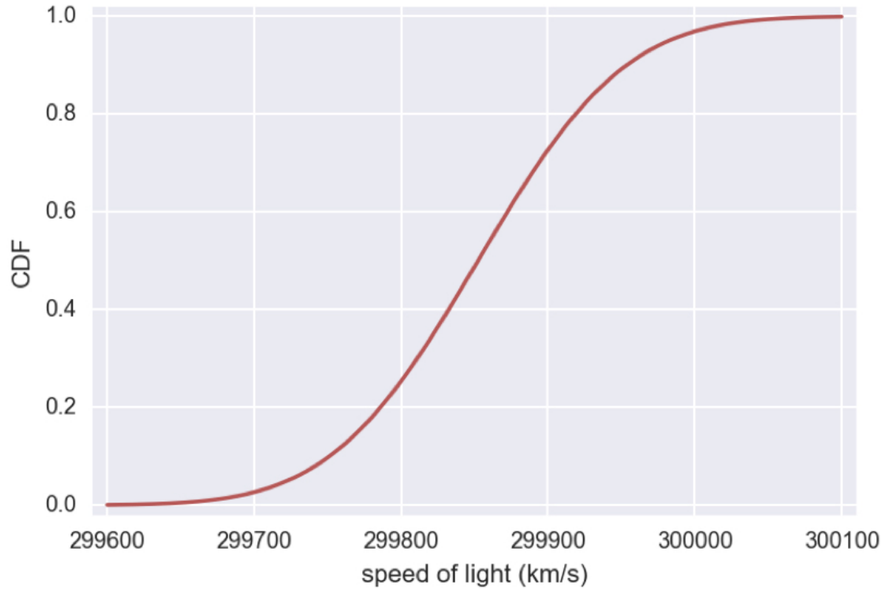
Normal PDF


0.006
0.005
0.004
0.003
0.002
0.001
0.000
PDF
299600
299700
299800
299900
300000
300100
speed of light (km/s)

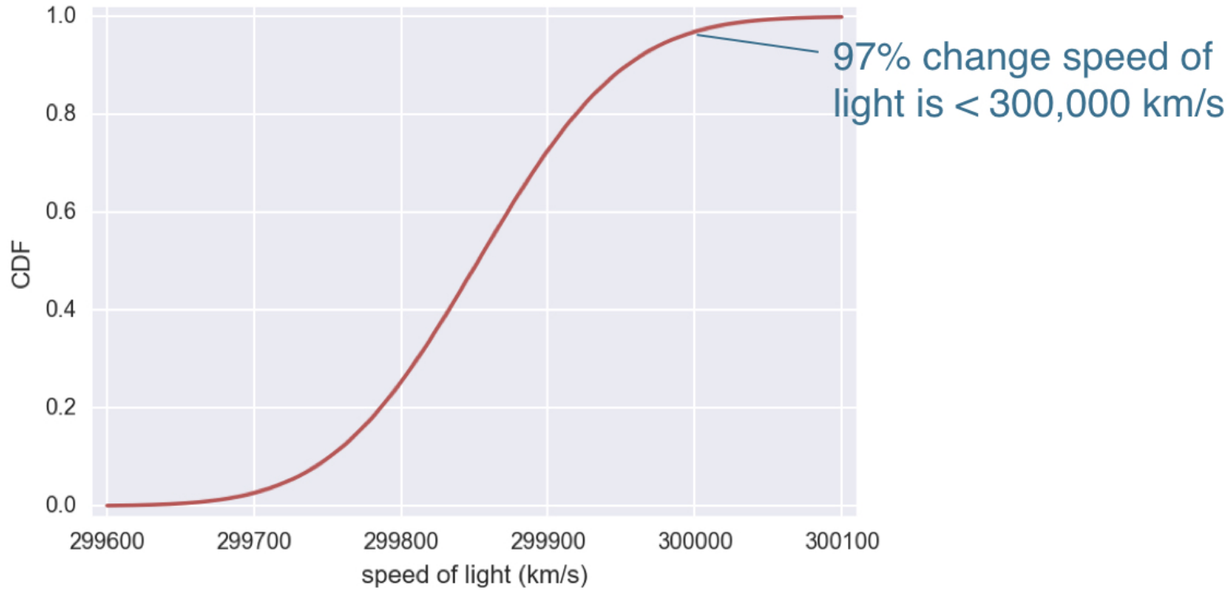
Normal PDF


0.006
0.005
0.004
0.003
0.002
0.001
0.000
PDF
299600
299700
299800
299900
300000
300100
speed of light (km/s)
3% of total area
under PDF

Normal CDF


1.0
0.8
0.6
0.4
0.2
0.0
CDF
299600
299700
299800
299900
300000
300100
speed of light (km/s)

Normal CDF


1.0
0.8
0.6
0.4
0.2
0.0
CDF
299600
299700
299800
299900
300000
300100
speed of light (km/s)
97% change speed of
light is < 300,000 km/s

Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

Introduction to the Normal distribution

STATISTICAL THINKING IN PYTHON (PART 1)

Justin Bois

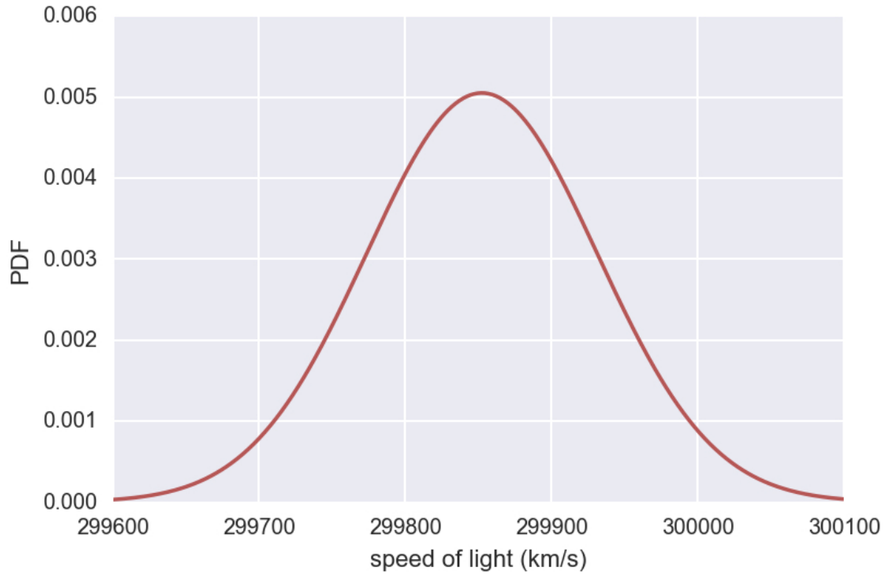
Lecturer at the California Institute of Technology

datacamp

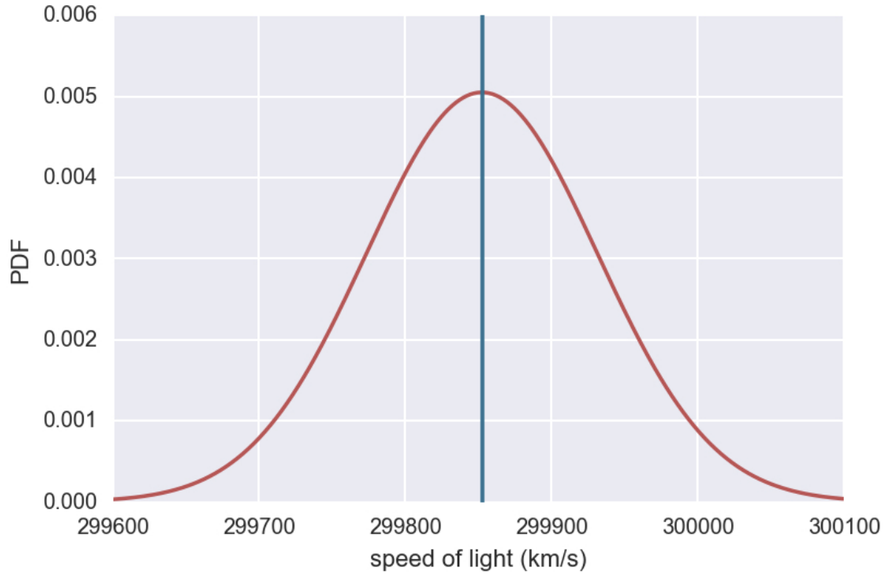
Normal distribution

- Describes a continuous variable whose PDF has a single symmetric peak.

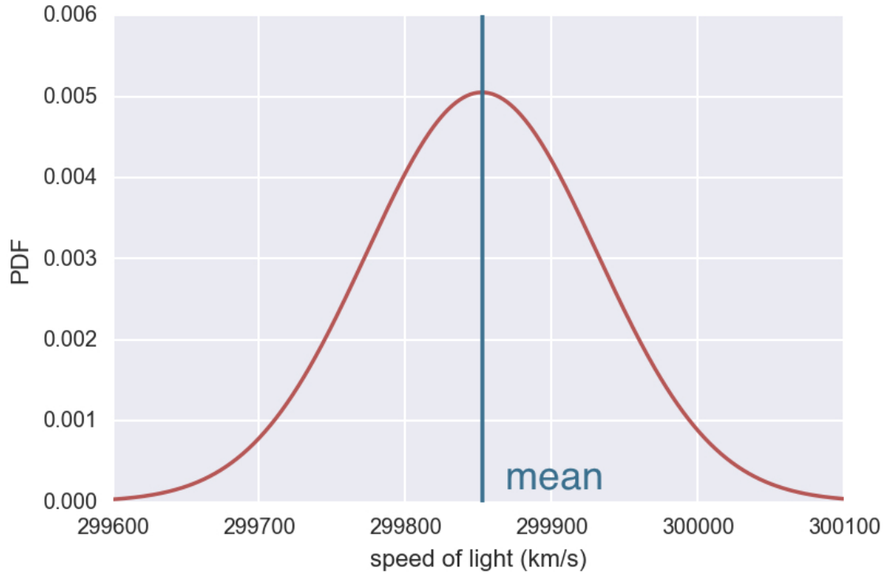
Normal distribution



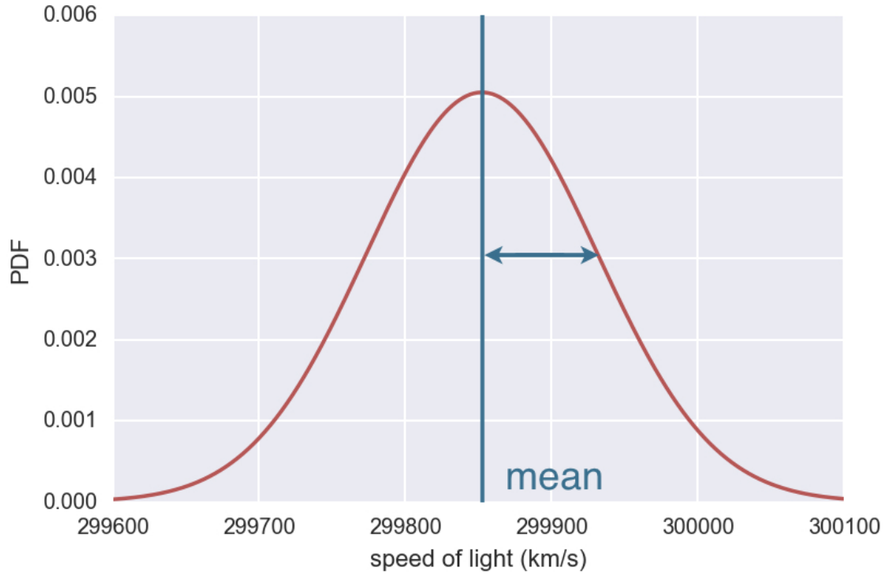
Normal distribution



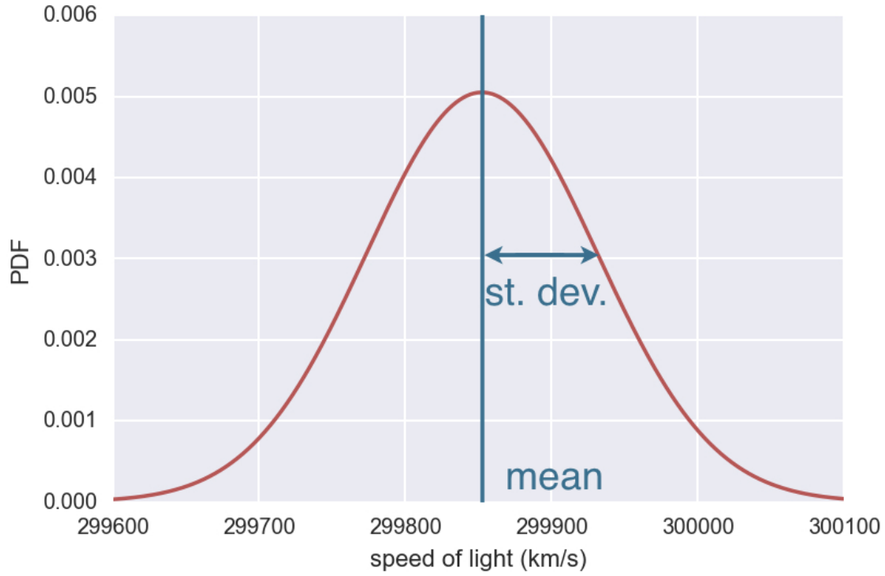
Normal distribution



Normal distribution



Normal distribution

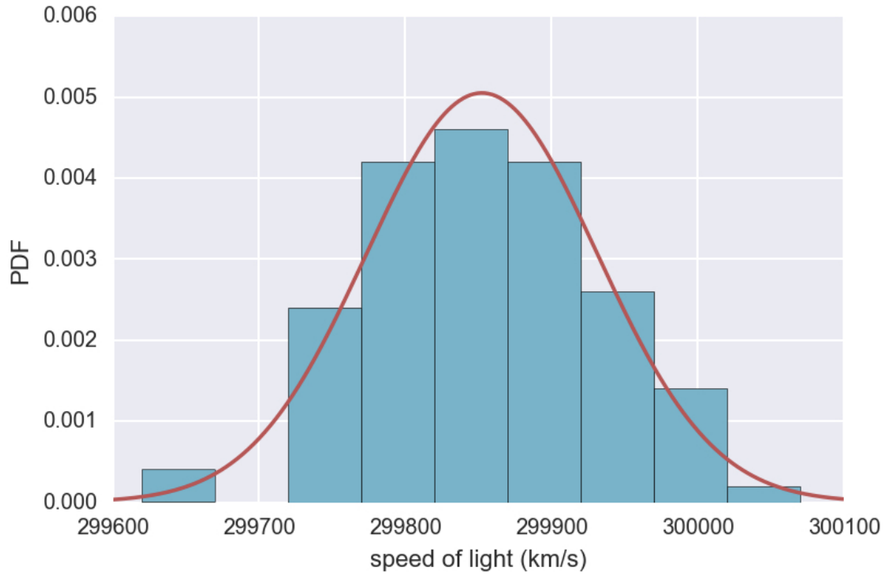


Parameter		Calculated from data
mean of a Normal distribution	≠	mean computed from data

Do not confuse these with the mean and standard deviation that we computed directly from the data when doing exploratory data analysis. YES! that nomenclature is confusing, but it is important to keep straight.

Parameter		Calculated from data
mean of a Normal distribution	≠	mean computed from data
st. dev. of a Normal distribution	≠	standard deviation computed from data

Comparing data to a Normal PDF



Data: Michelson, 1880

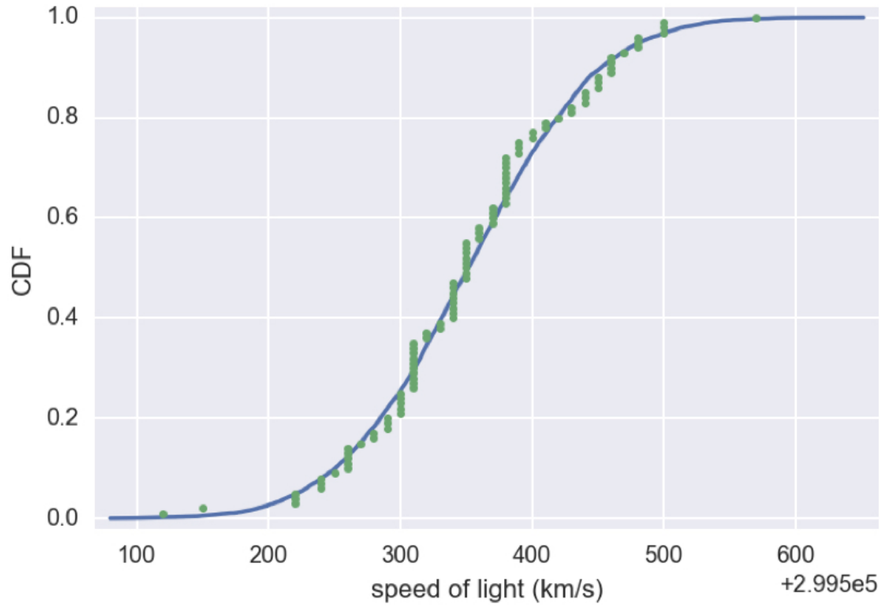
Checking Normality of Michelson data

```
import numpy as np
mean = np.mean(michelson_speed_of_light)
std = np.std(michelson_speed_of_light)
samples = np.random.normal(mean, std, size=10000)
x, y = ecdf(michelson_speed_of_light)
x_theor, y_theor = ecdf(samples)
```

Checking Normality of Michelson data

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
_ = plt.plot(x_theor, y_theor)
_ = plt.plot(x, y, marker='.', linestyle='none')
_ = plt.xlabel('speed of light (km/s)')
_ = plt.ylabel('CDF')
plt.show()
```

Checking Normality of Michelson data



Data: Michelson, 1880

Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

The Normal distribution: Properties and warnings

STATISTICAL THINKING IN PYTHON (PART 1)

Justin Bois

Lecturer at the California Institute of Technology

datacamp

Image: Deutsche Bundesbank

Image: Deutsche Bundesbank

The Gaussian distribution

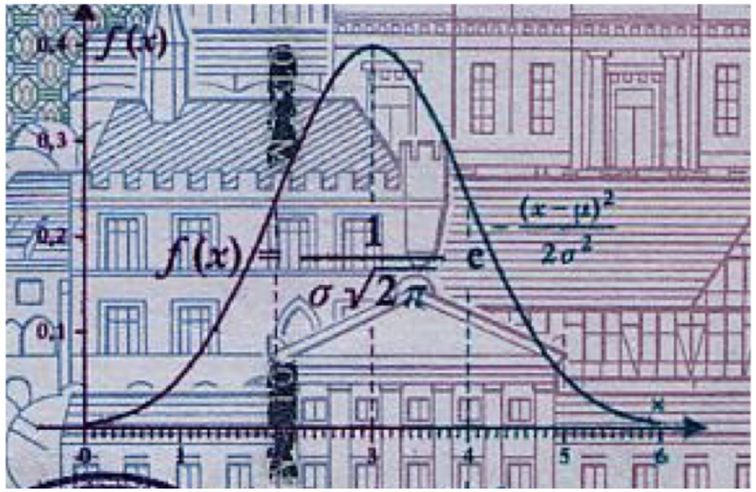
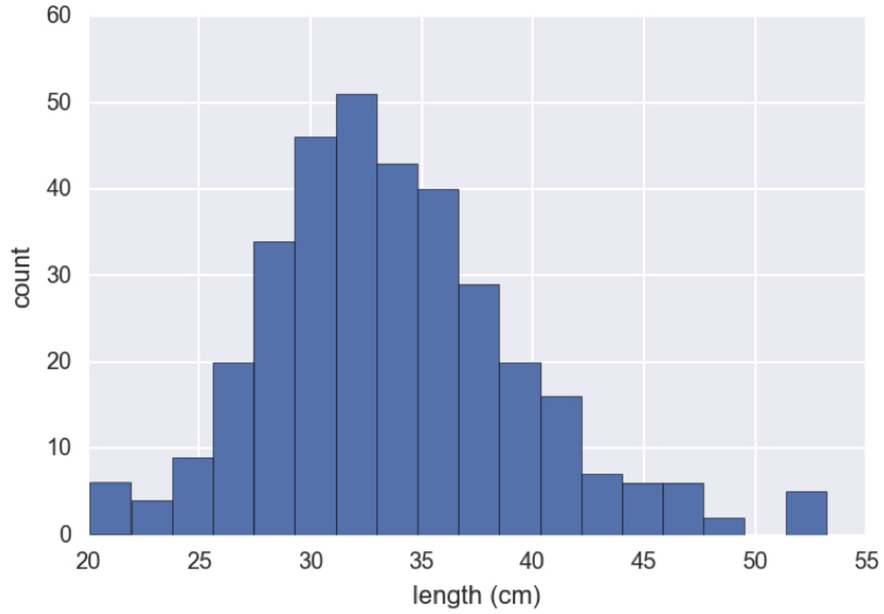


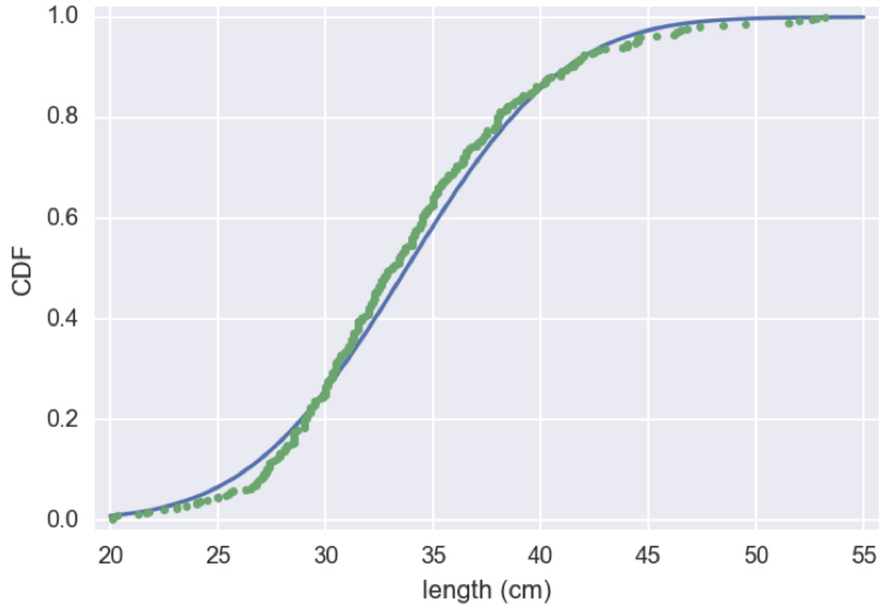
Image: Deutsche Bundesbank

Length of MA large mouth bass



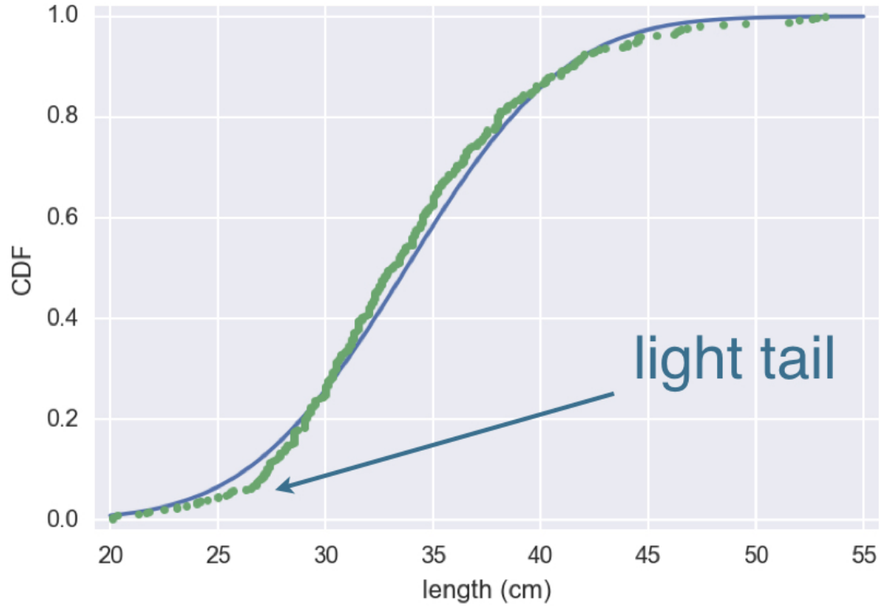
Source: Mass. Dept. of Environmental Protection

Length of MA large mouth bass



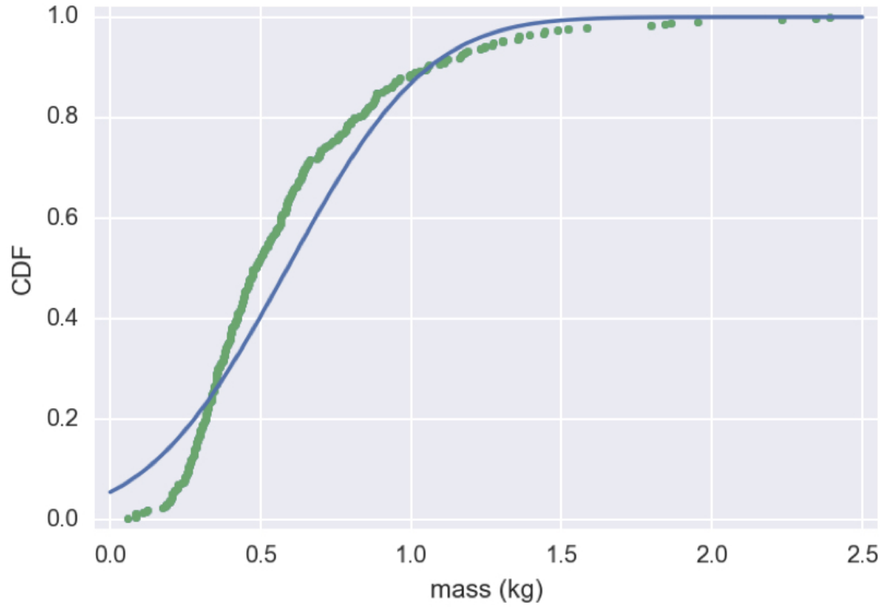
Source: Mass. Dept. of Environmental Protection

Length of MA large mouth bass



Source: Mass. Dept. of Environmental Protection

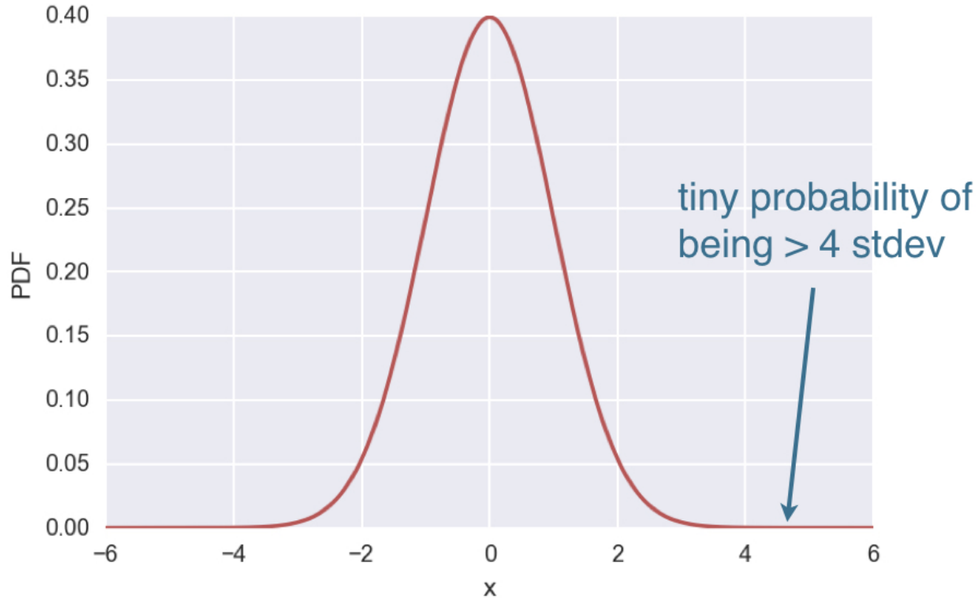
Mass of MA large mouth bass



Source: Mass. Dept. of Environmental Protection

Light tails of the Normal distribution

Light tails of the Normal distribution



Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

The Exponential distribution

STATISTICAL THINKING IN PYTHON (PART 1)

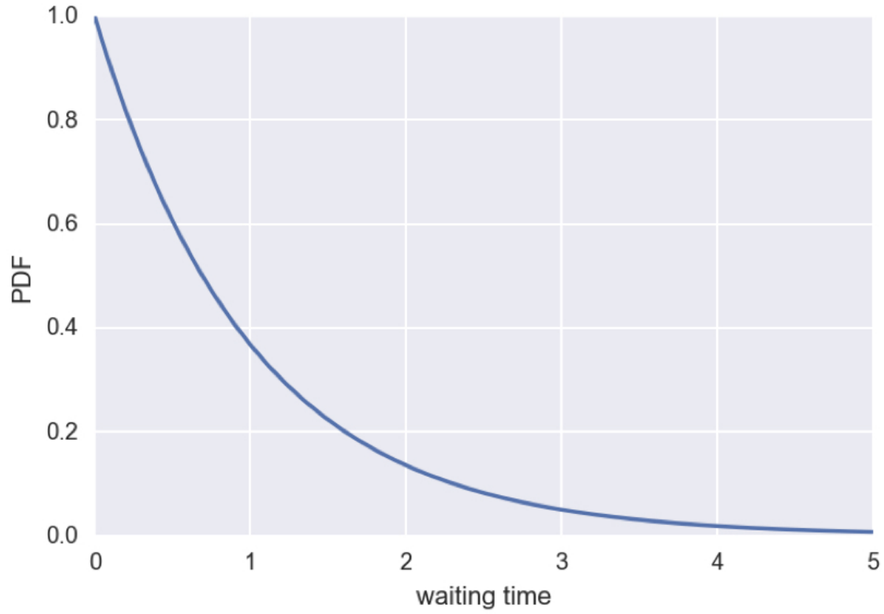
Justin Bois

Lecturer at the California Institute of Technology

The Exponential distribution

- The waiting time between arrivals of a Poisson process is Exponentially distributed

The Exponential PDF



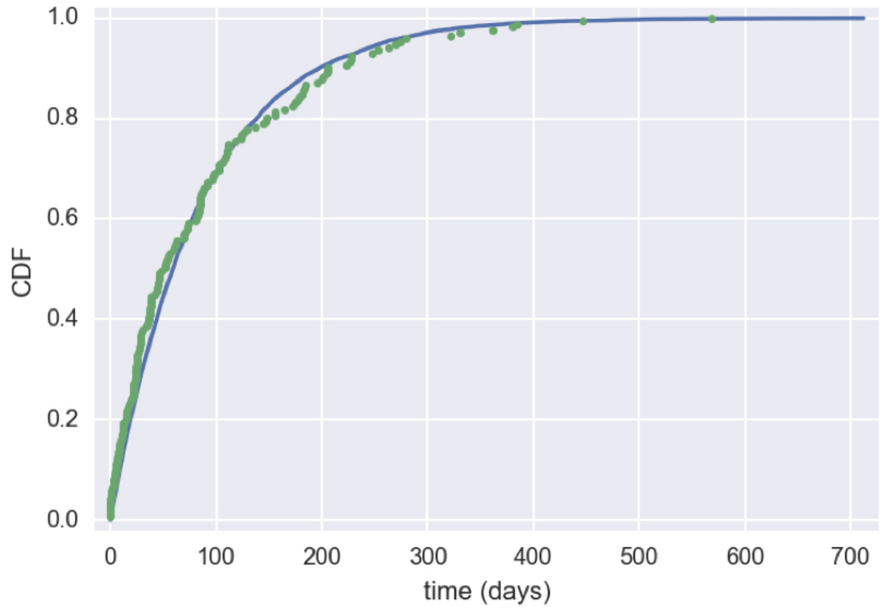
Possible Poisson process

- Nuclear incidents:
  - Timing of one is independent of all others

Exponential inter-incident times

```
mean = np.mean(inter_times)
samples = np.random.exponential(mean, size=10000)
x, y = ecdf(inter_times)
x_theor, y_theor = ecdf(samples)
_ = plt.plot(x_theor, y_theor)
_ = plt.plot(x, y, marker='.', linestyle='none')
_ = plt.xlabel('time (days)')
_ = plt.ylabel('CDF')
plt.show()
```

Exponential inter-incident times



We can compute and plot the CDF we would expect based on the mean time between incidents and overlay that with the ECDF from the real data. We take our usual approach where we draw many samples out of the Exponential distribution, using the mean inter-incident time as the parameter. We make the plot and label the axes.

We see that it is close to being Exponentially distributed, indicating the nuclear incidents can indeed be modeled as a Poisson process. The Exponential and Normal are just two of many examples of continuous distributions. Importantly, in many cases you can just simulate your story to get the CDF. Remember, you have the power of a computer. If you can simulate a story, you can get its distribution!

Data Source: Wheatley, Sovacool, Sornette, Nuclear Events Database

Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)

Final thoughts

STATISTICAL THINKING IN PYTHON (PART 1)

Justin Bois

Lecturer at the California Institute of Technology

You now can...

- Construct (beautiful) instructive plots
- Compute informative summary statistics
- Use hacker statistics
- Think probabilistically

In the sequel, you will...

- Estimate parameter values
- Perform linear regressions
- Compute confidence intervals
- Perform hypothesis tests

Let's practice!

STATISTICAL THINKING IN PYTHON (PART 1)