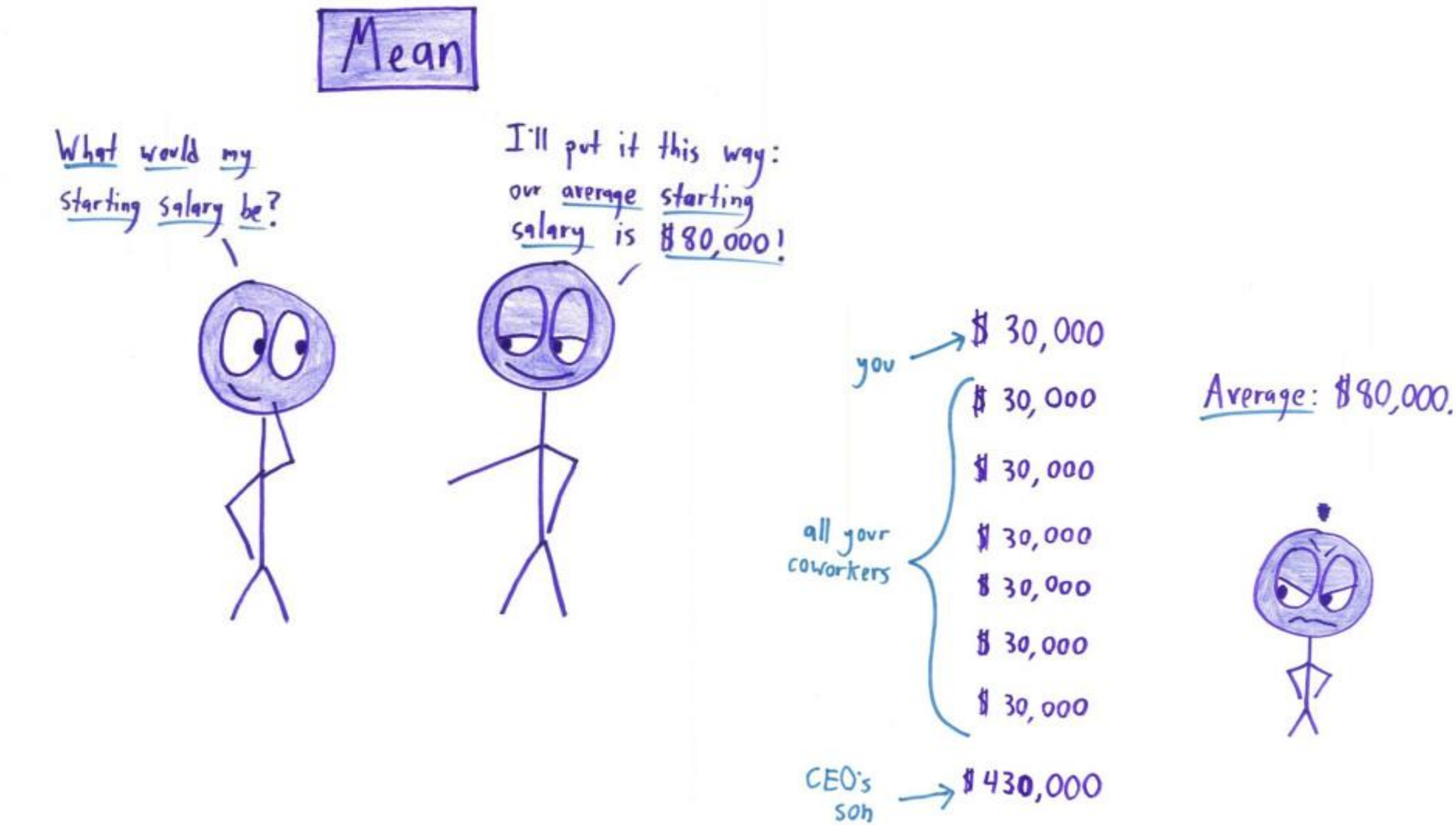
Lecture 4 – Statistics II



February 3rd, 2026

Spring 2026

Today's Learning Outcomes

1. Be able to explain, in plain language, what the standard deviation formula is actually measuring

2. Be able to compile measures of central tendency and variance into a data frame in Python

Summarizing Data

- Since we want large sample sizes of variable data there’s a need to summarize the data (i.e. reduce it to a few easily understood numbers)

- Variance/Uncertainty
  - Standard deviation
  - Confidence intervals
  - Max/Min values

Range of possible values/estimates

- Shape of a distribution
  - Mean
  - Mode
  - Percentiles
    - Median
    - Quartiles
    - Quintiles

Which values are most common relative to other values

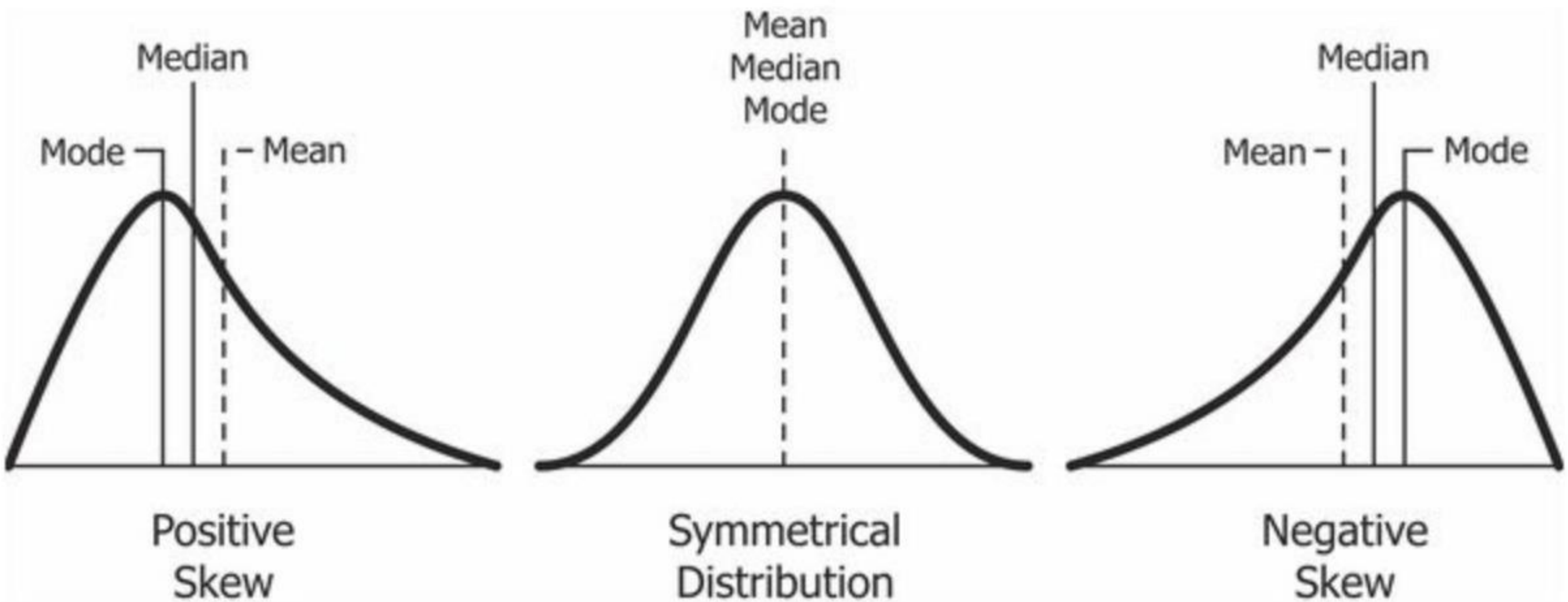
Shape of Distributions

- Mean, median, mode ← measures of central tendency
  - i.e. where the center of sample values is
- If the goal is to actually find the middle of a sample/distribution the median is the most robust
- mean() and median()
  - No base command for mode (have to make a function)

```
a = [1,2,2,3,3,3]
# to calculate mean and median, we can simply use numpy
import numpy as np
a_mean = np.mean(a)
a_median = np.median(a)
```

```
# to calculate mode, we will need to use pandas
import pandas as pd
s = pd.Series([1,2,2,3,3])
print(s.mode())
```

- The mean value is strongly affected by outlier values (i.e. extremes) and mode is vulnerable/uninformative when values are relatively even
  - Mean shifts toward the long tail of a distribution

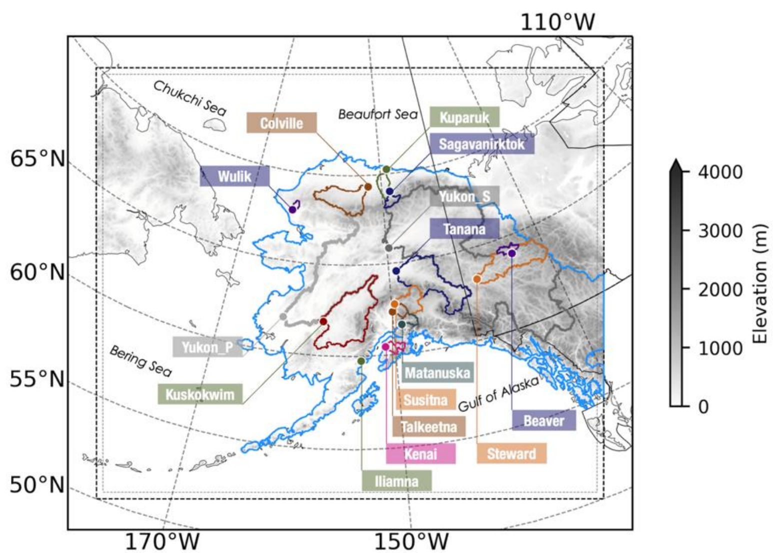


Case study: how would you report your results?

	RASM	CTSM
Iliamna	**0.49**	0.32
Wulik	0.17	0.25
Beaver	0.51	0.51
Kuparuk	0.13	0.35
Sagavanirktok	**0.62**	0.53
Matanuska	0.47	0.59
Talkeetna	**0.65**	0.55
Kenai	**0.50**	0.43
Steward	**0.71**	0.64
Susitna	**0.70**	0.61
Colville	0.00	0.47
Tanana	**0.61**	0.56
Kuskokwim	**0.22**	0.03
Yukon_S	**0.58**	0.50
Yukon_P	**0.60**	0.50
median	**0.51**	**0.5**
mean	**0.464**	**0.456**

Left table shows the Nash-Sutcliffe Efficiency (**NSE**) of two model simulations(**RASM** *versus* **CTSM**) against USGS observations for 15 major river basins across.

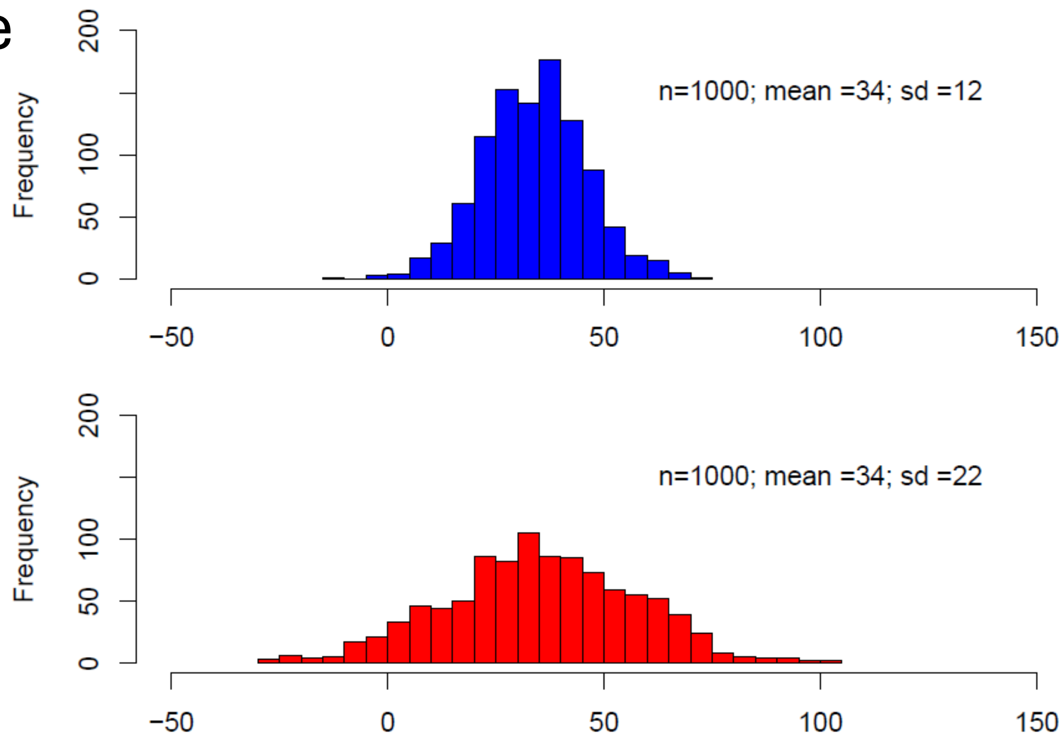
1. Higher **NSE** values mean better performance
2. The basins are ordered from small basins to large basins (from top to bottom)



Source: https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2022WR032204

Measuring Variance of Samples

- Point values alone only tell us so much about data
- We need to measure the spread of values as well
  - Percentiles
  - Standard deviation
  - Max/min

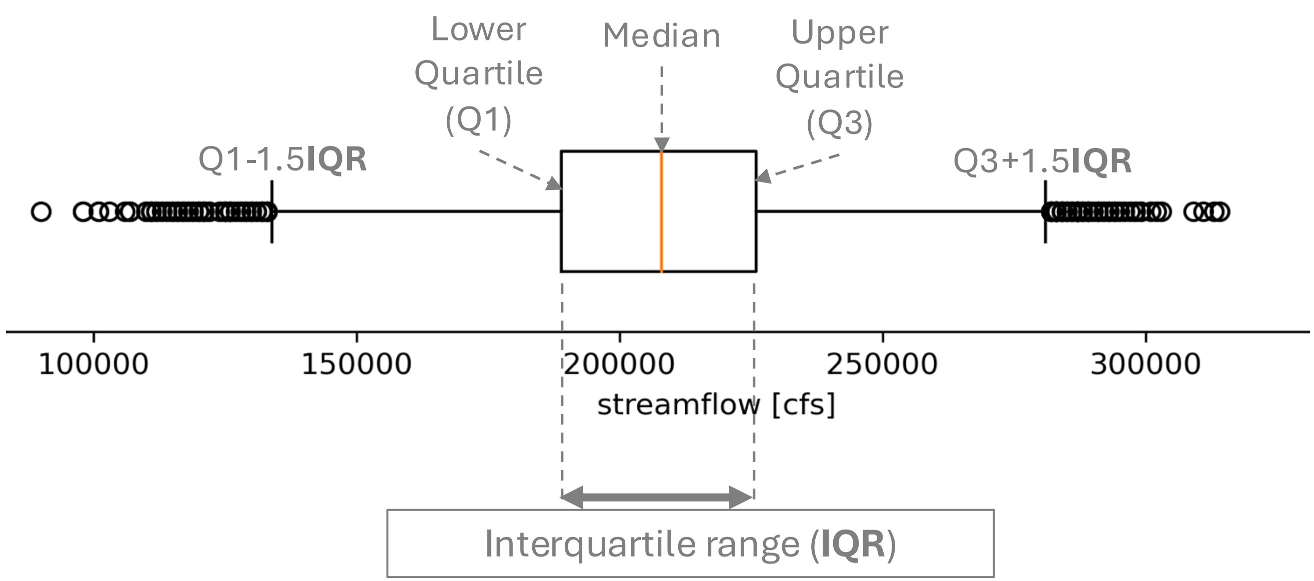


Percentiles

- Percentiles are the values of a distribution that x% of the distribution is less than or equal to
  - The median is the 50th percentile

- Common percentiles are 25th, 50th, and 75th
  - Interquartile range of boxplots are the values between the 25th and 75th percentiles

boxplot



Standard Deviation

Python function

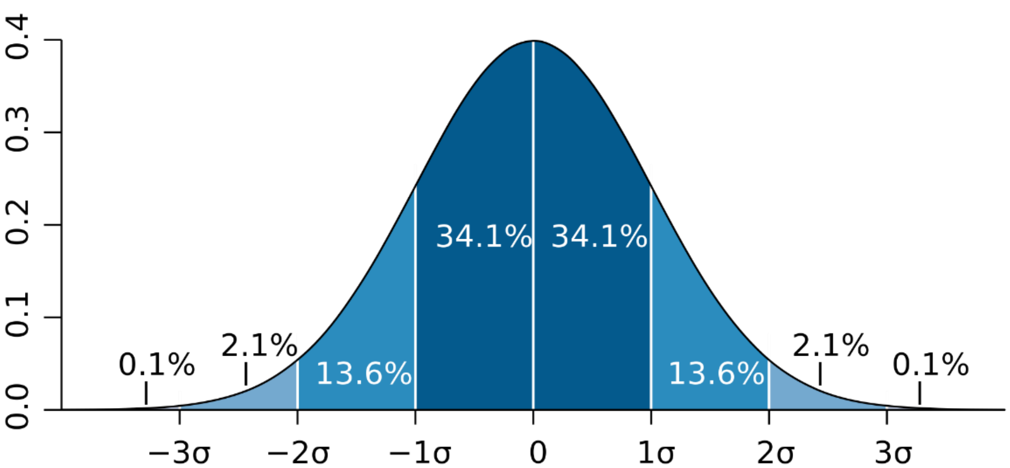
```
np.std()
```

$$\sigma = \sqrt{(\frac{\Sigma(x_i - \mu)^2}{N-1})}$$

σ = standard deviation
x_i = a data point value
μ = mean value of the data
N = total number of data points
Σ = sum operator

Standard deviation reflects the spread of values assuming the data being sampled is from a **normal distribution**

The 2nd power means big deviations from the mean are weighted more heavily (i.e. they indicate a flatter and broader normal distribution with long tails)



Increasing sample size (N) decreases the calculated SD as the chance of values being poor representative of the distribution decreases [i.e. less randomness]

Summarizing Data

- As with any summary these numbers flatten out variation

- Always useful to plot your actual data and make sure they are appropriate and actually represent what you think they do!

- Open Github Codespace in your homework repo

- Go to “CourseMaterials/coding_modules/” and double click “lec4_inclass_practice_basic_statistics.ipynb”

- Four columns of 1000 values, each sampled randomly from a different distribution

1. Load "Lect4_Data.csv" file into Python

```
import pandas as pd
pd.read_csv()
```

2. Create a histogram for each of the four datasets on one plot

```
import matplotlib.pyplot as plt
plt.hist()
```

3. Make a data frame containing the mean, 25^{th} percentile, median, 75^{th} percentile, and standard deviation of each of the four distributions

```
pd.DataFrame()
```

- If we wanted to get the quintiles (0,20,40,60,80,100) of one of these how would you do so?
  - Just conceptually for now, you don't have to try and write a Python code

Measuring Uncertainty

- Acknowledge that we have only a sample estimate (not the population) so if we were to resample we would almost certainly get a slightly different answer

- Confidence intervals (CI) are ways of expressing the uncertainty in an estimated value for a sampling method
  - Can be derived by assuming/fitting a distribution to data and then the range of values expected by that distribution
    - Ex. CI of Correlation coefficient in Python `pearsonr(x1, x2)`
  - Resampling techniques (bootstrap/jackknife/Monte Carlo)

* `Pearsonr` needs to be imported from scipy package

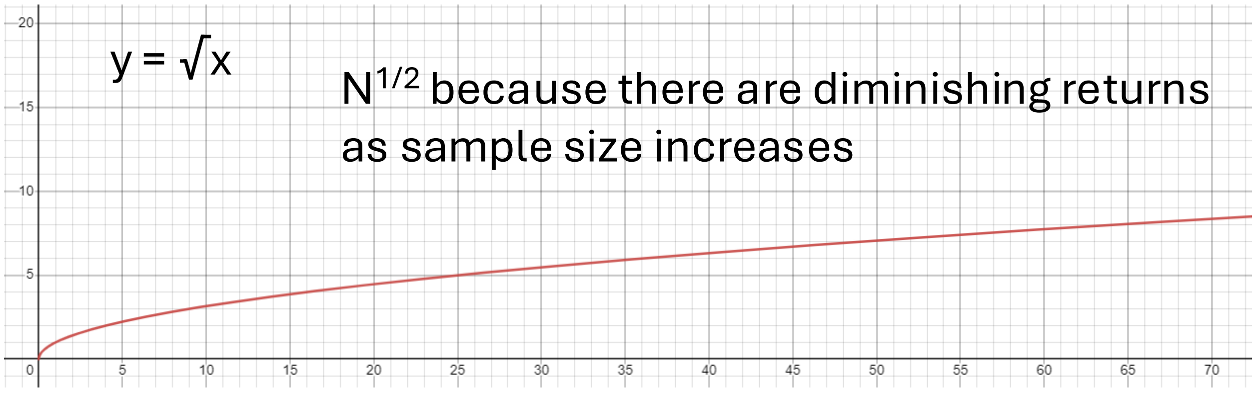
```
from scipy.stats import pearsonr
```

Standard Error (SE)

- Standard error (SE) = (standard deviation of the data)/$(N)^{1/2}$ ← $x^{1/2} = \sqrt{x}$

- Measure of uncertainty in the observed mean based on sample size
  - As sample size increases there is **more certainty** in the estimated mean value

Is our estimate likely to change very much with 10,000 versus 20,000 values?

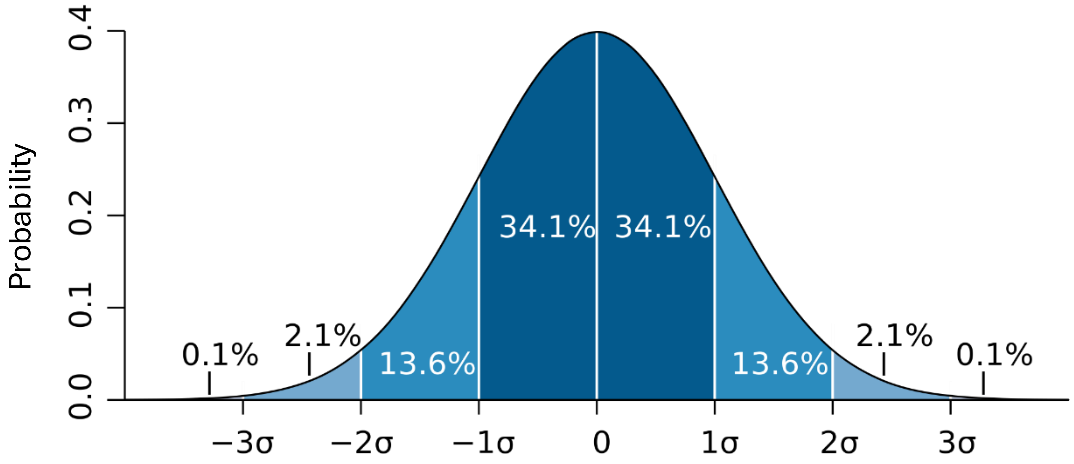


Uncertainty in Estimates

- Reported confidence intervals are often 95% CI
  - Tied to old practice of using p-values with $p < 0.05$ being considered statistically significant

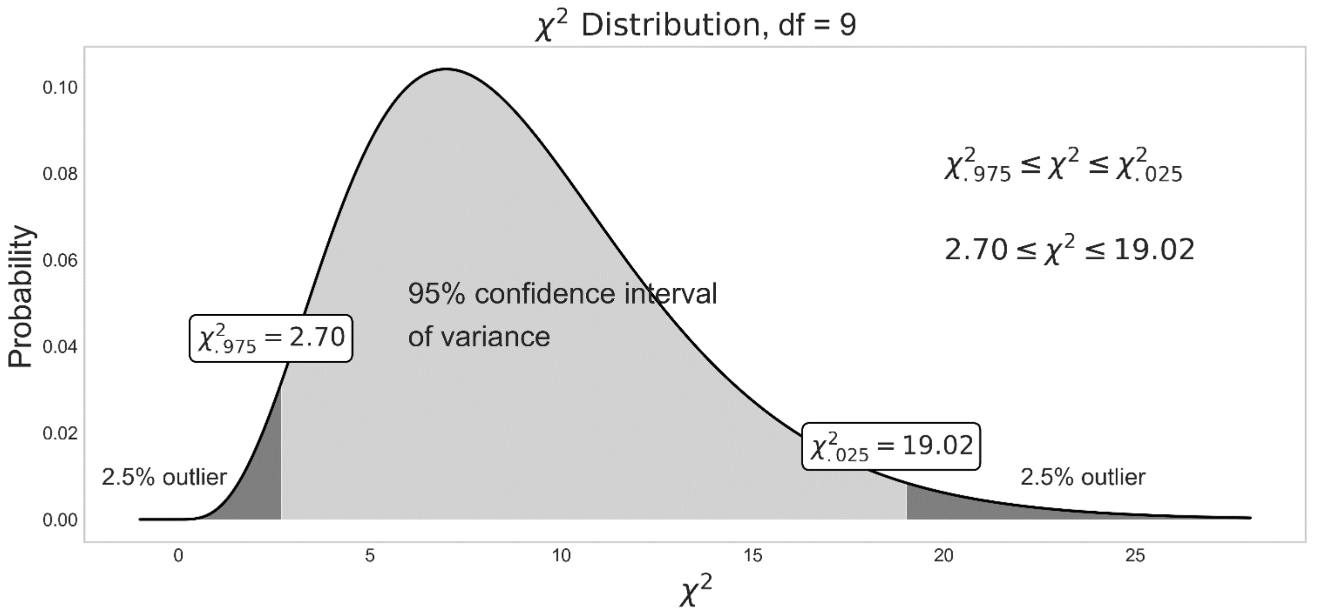
- P-value of 0.05 ≈ 5% chance of result by random chance

- ≈ 2 standard deviations for a normal distribution
  - Chosen totally arbitrarily as the standard



Uncertainty in Estimates

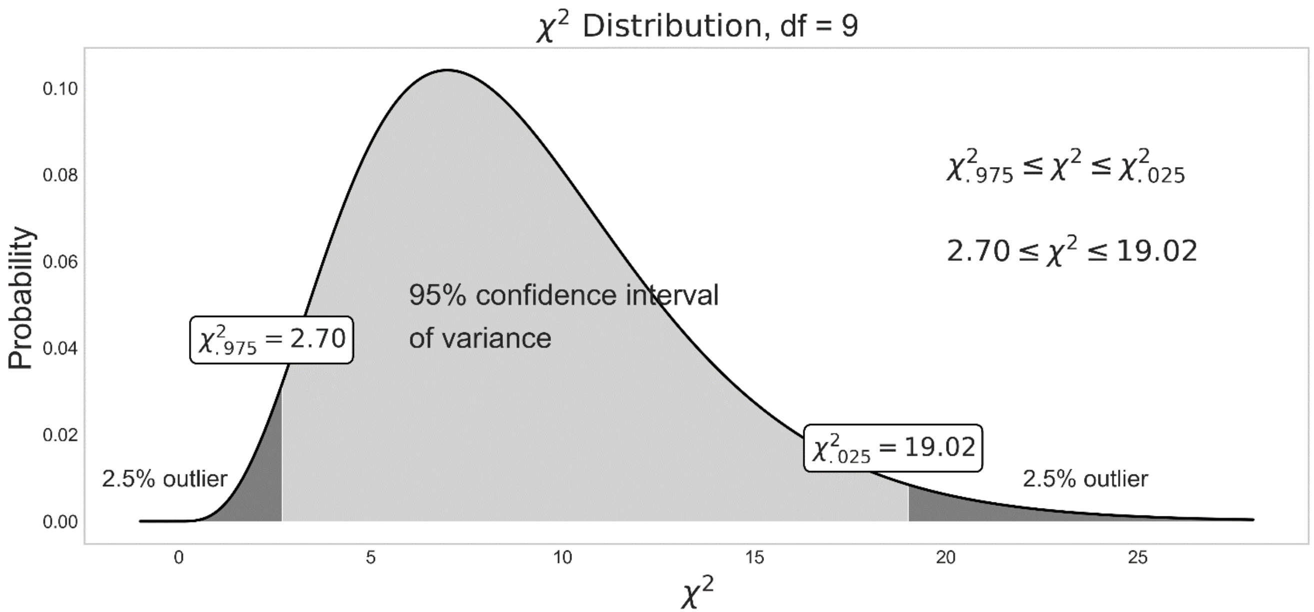
- Lots of data not normal so cannot use 2SD as our standard in many cases (or at least that range is not what most people will think of it as)
  - But 95% of the range of values observed IS useful!
  - Range of values between the 5th and 95th percentiles



Uncertainty in Estimates

- Can use same approach as making quintiles to make 95% (or any other range)

```
NinetyFiveCI = [np.percentile(x,2.5),np.percentile(x,97.5)]
```



Today’s Learning Outcomes

1. Be able to explain, in plain language, what the standard deviation formula is actually measuring

2. Be able to compile measures of central tendency and variance into a data frame in Python