Who is Bayes? What is Bayes?

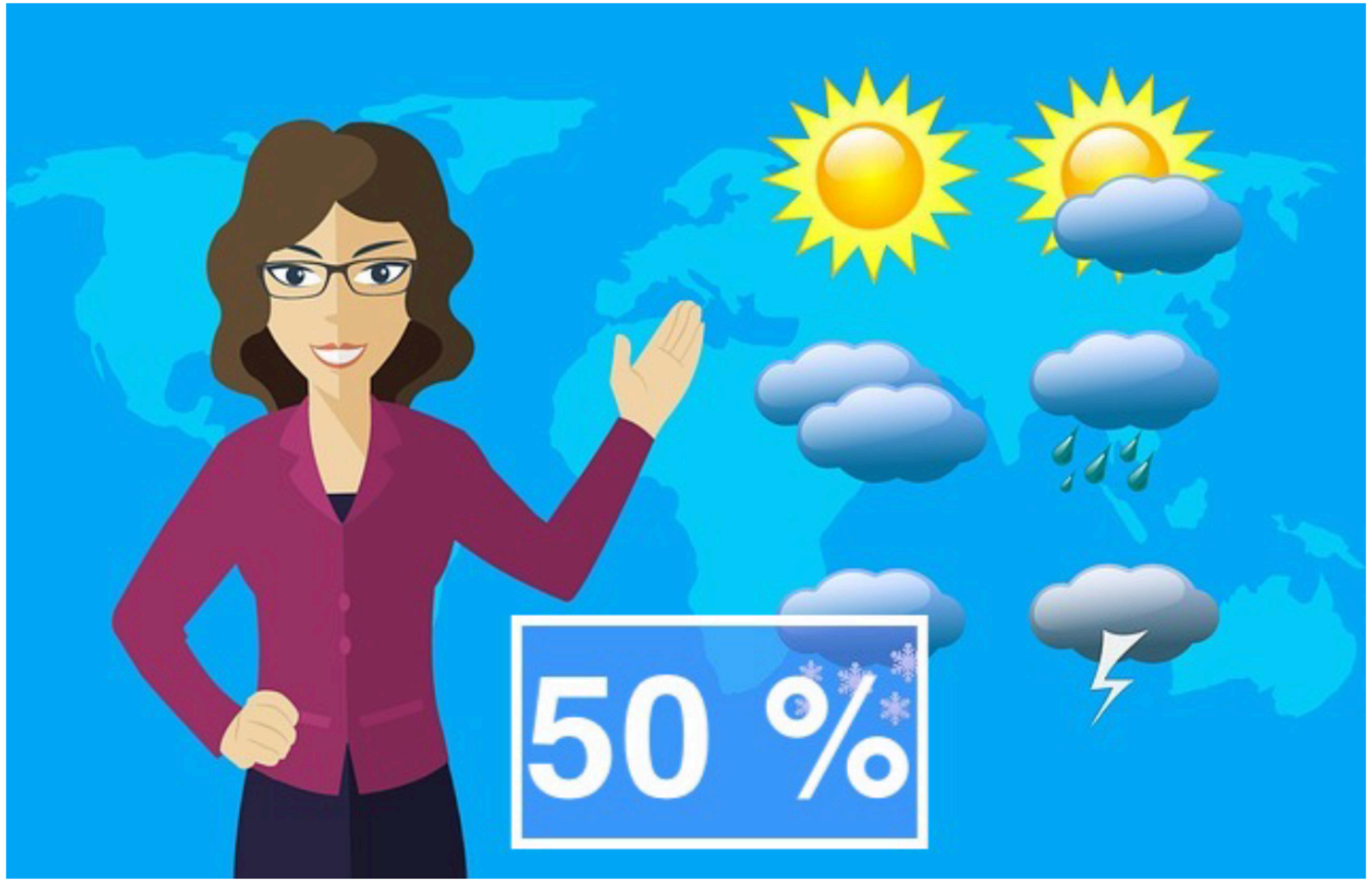
BAYESIAN DATA ANALYSIS IN PYTHON

Michal Oleszak
Machine Learning Engineer

Who is Bayes?

[1] Public Domain, https://commons.wikimedia.org/w/index.php?curid=14532025

Should you take your umbrella?



What is Bayes?

- Bayesian inference means updating one's belief about something as the new information becomes available.

What is Bayes?

- Bayesian inference means updating one's belief about something as the new information becomes available.
- It is quite different from the classical approach.

	Frequentist (classical) approach	Bayesian approach
probability		
parameters		

What is Bayes?

- Bayesian inference means updating one's belief about something as the new information becomes available.
- It is quite different from the classical approach.

	Frequentist (classical) approach	Bayesian approach
probability	proportion of outcomes	degree of belief
parameters		

What is Bayes?

- Bayesian inference means updating one's belief about something as the new information becomes available.
- It is quite different from the classical approach.

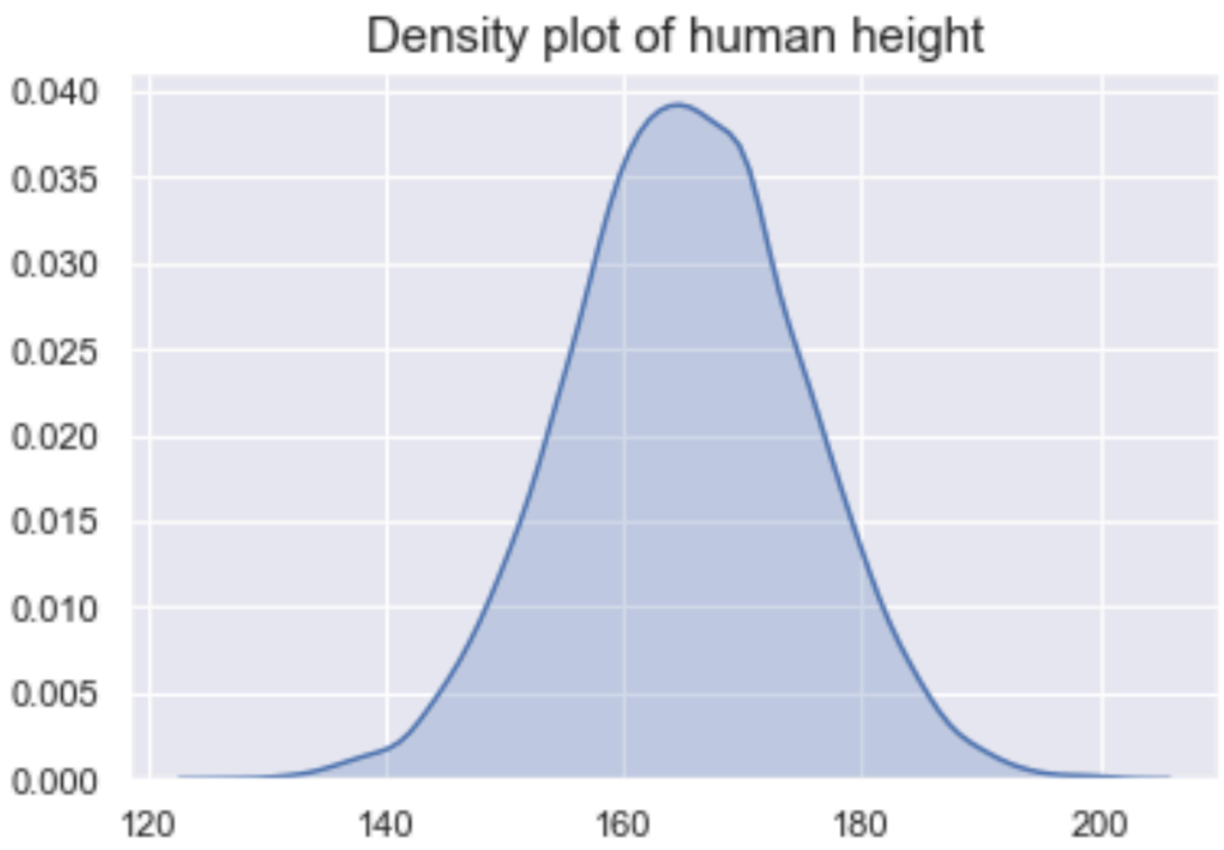
	Frequentist (classical) approach	Bayesian approach
probability	proportion of outcomes	degree of belief
parameters	fixed values	random variables

It pays to go Bayes!

- Natural handling of uncertainty (because parameters have distributions!).
- Possibility to include expert opinion or domain knowledge in the model (because probability means the degree of belief!).
- No need to rely on fixed constants such as p-values.
- Statistically correct even with little data.
- Often coincides with frequentist results, but offers more flexibility to build custom models.

Probability distributions

- A distribution of a random variable specifies what values this variable can take, and with what probabilities.
- Can be discrete (finite set of possible values) or continuous (infinitely many possible values)
- Continuous distributions can be visualized on a density plot.



Distributions in Python

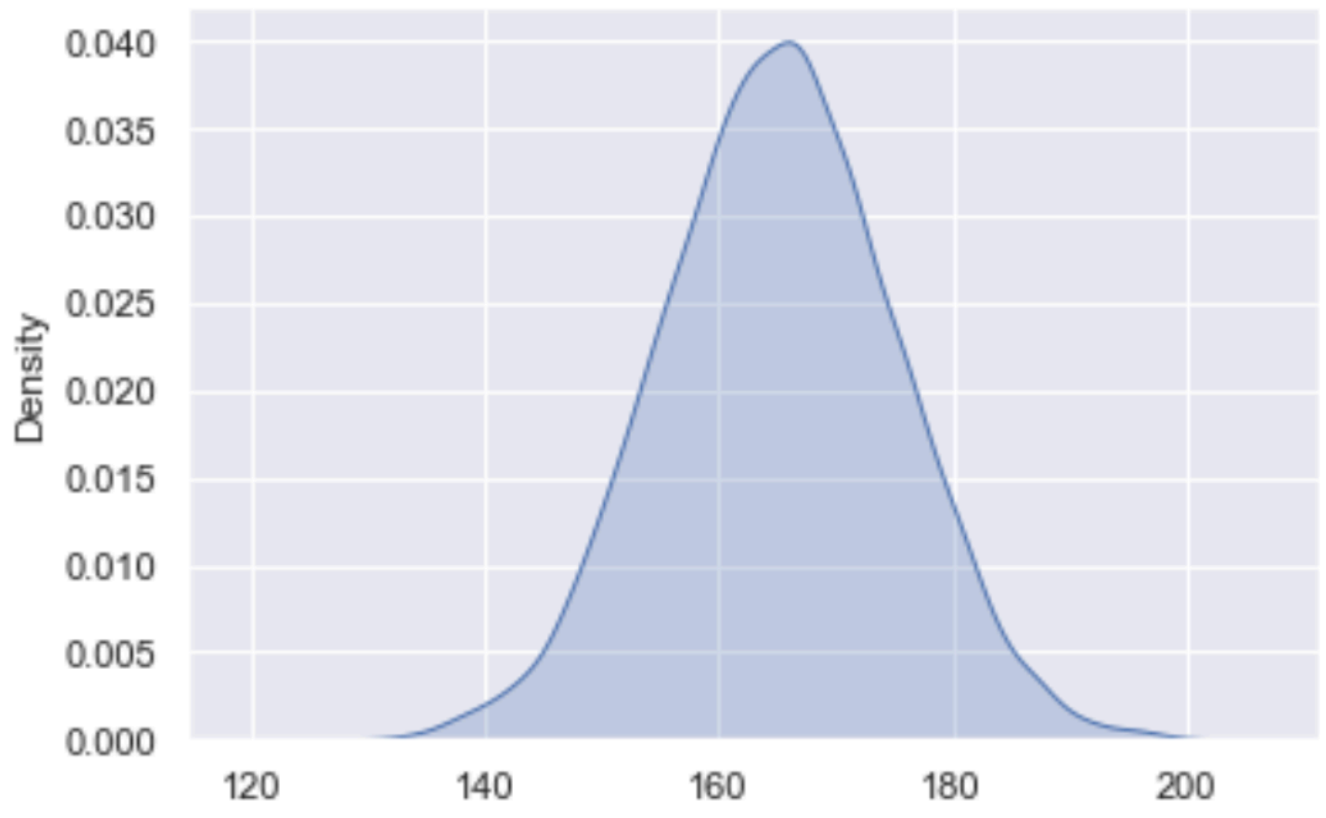
```
print(draws)
```

```
[146.58686154393, 159.40688614250, ..., ]
```

```
print(len(draws))
```

```
10000
```

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.kdeplot(draws, shade=True)
plt.show()
```



Probability and Bayes' Theorem

BAYESIAN DATA ANALYSIS IN PYTHON

Michal Oleszak
Machine Learning Engineer

Probability theory

- Statement of uncertainty.
- A number between 0 and 1.
  - P = 0 → impossible
  - P = 1 → certain
  - P = 0.5 → 50/50 chance

- P(rain tomorrow) = 0.75 → 75% chance of rain tomorrow

Probability rules

Sum rule

- Probability of A **or** B (independent events)
- OR = addition
- Probability of rolling 2 or 4 with a die

P(2 or 4) = 1/6 + 1/6 = 0.33333... = 33.3%

Product rule

- Probability of A **and** B (independent events)
- AND = multiplication
- Probability of rolling 2 and then 4 with a die

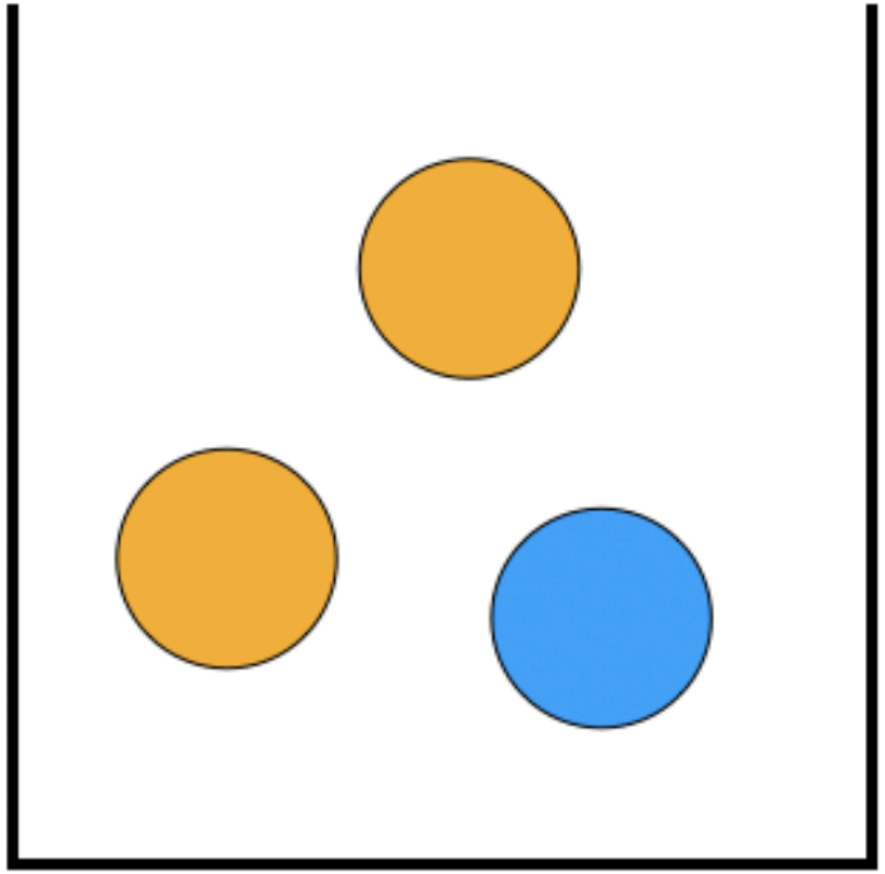
P(2 and 4) = 1/6 * 1/6 = 0.02777... = 2.8%

Conditional probability

- Probability of some event occurring, **given that** some other event has occurred.
- P(A | B)

- P(orange) = 2/3 → unconditional
- P(blue) = 1/3 → unconditional

- P(blue | orange) = 1/2 → conditional
- P(orange | blue) = 1 → conditional



Bayes' Theorem

- A way to calculate conditional probability when we know some other probabilities.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Bayes' Theorem

- A way to calculate conditional probability when we know some other probabilities.

$$P(\text{accident}|\text{slippery}) = \frac{P(\text{slippery}|\text{accident}) * P(\text{accident})}{P(\text{slippery})}$$

```
road_conditions.head()
```

```
     accident   slippery
0       False       True
1        True       True
2       False      False
3       False      False
4       False      False
```

Bayes' Theorem in practice

$$P(\text{accident}|\text{slippery}) = \frac{P(\text{slippery}|\text{accident}) * P(\text{accident})}{P(\text{slippery})}$$

```
# Unconditional probability of an accident
p_accident = road_conditions["accident"].mean()  # 0.0625

# Unconditional probability of the road being slippery
p_slippery = road_conditions["slippery"].mean()  # 0.0892

# Probability of the road being slippery given there is an accident
p_slippery_given_accident = road_conditions.loc[road_conditions["accident"]]["slippery"].mean()  # 0.7142

# Probability of an accident given the road is slippery
p_accident_given_slippery = p_slippery_given_accident * p_accident / p_slippery  # 0.5
```

Tasting the Bayes

BAYESIAN DATA ANALYSIS IN PYTHON

Michal Oleszak
Machine Learning Engineer

Binomial distribution

- A discrete distribution, can only take one of two values:
  - Success (1)
  - Failure (0)
- One parameter: probability of success.
- Task: given a list of draws (successes and failures), estimate the probability of success.

Binomial distribution in Python

Number of successes in 100 trials:

```
import numpy as np
np.random.binomial(100, 0.5)
```

```
51
```

```
np.random.binomial(100, 0.5)
```

```
44
```

Get draws from a binomial:

```
import numpy as np
np.random.binomial(1, 0.5, size=5)
```

```
array([1, 0, 0, 1, 1])
```

Heads probability

- `get_heads_prob()` - a custom function
- input: a list of coin tosses
- output: a list, the distribution of the probability of success

```
import numpy as np
tosses = np.random.binomial(1, 0.5, size=1000)
print(tosses)
```

```
[1 0 0 0 1 1 0 1 1 ... ]
```

Heads probability

```python
heads_prob = get_heads_prob(tosses)
print(heads_prob)
```

```
[0.47815295 0.51679212 0.51684779 ... ]
```

```python
import matplotlib.pyplot as plt
import seaborn as sns

sns.kdeplot(heads_prob,
            shade=True,
            label="heads probabilty")
plt.show()
```