# Bayesian Statistics

M. Fuat Kına

# What is Bayes?

- Thomas Bayes was an eighteenth-century English statistician.
- The example raining
  - On TV, you see a forecast predicting a 50% chance of rain. You start to consider taking the umbrella.
- Two paradigmatic differences between frequentist and bayesian approaches
  - The meaning of probability
    - Frequentists: the proportion of outcomes
    - Bayes: a degree of belief
  - Statistical parameters
    - Frequentists: fixed values
    - Bayes: random variables

Credits for:

https://app.datacamp.com/learn/courses/bayesian-data-analysis-in-python

# Bayesian parameters

- Have distributions
- Natural handling of uncertainity
- Depend on expert, domain-specific knowledge
- Do not depend on fixed constants such as p-values
- Provides more flexibility
- Works even with small data

# Probability

- P=0 impossible
- P=1 certain
- P(A) **and** P(B) means product
- P(A) **or** P(B) means sum
- P(A|B) means P(A) while B is given (**conditional**)
- $$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

# An example of sickness and test result

**What does the test say?**

A doctor suspects a disease in their patient, so they run a medical test. The test's manufacturer claims that 99% of sick patients test positive, while the doctor has observed that the test comes back positive in 2% of all cases. The suspected disease is quite rare: only 1 in 1000 people suffer from it.

The test result came back positive. What is the probability that the patient is indeed sick? We can use Bayes' Theorem to answer this question.

Let’s say being sick is A, and positive test result is B. We want to figure out $P(A|B)$

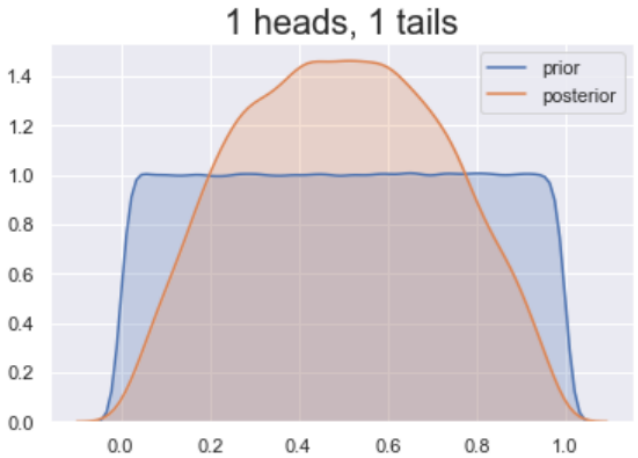
# Bayes' Theorem

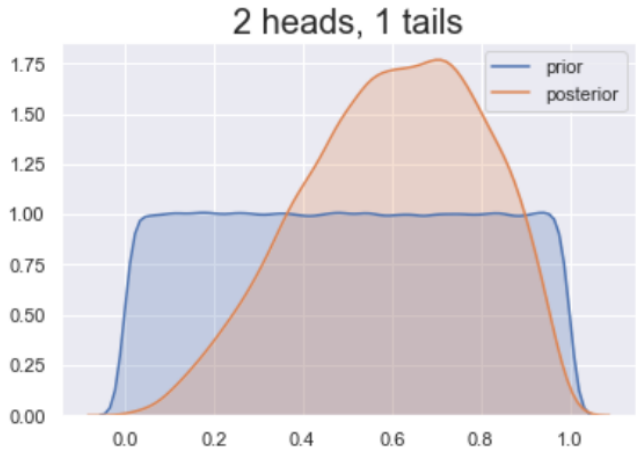
- Replace A with the parameters and B with the data

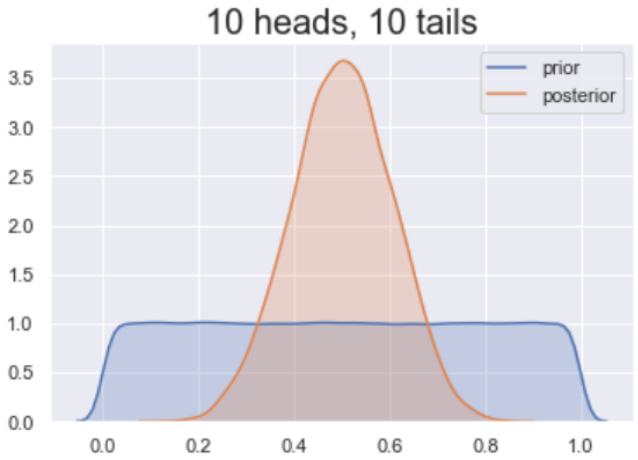
  - P(parameters|data) -> posterior distribution what we know about parameters after we see the data
  - P(parameters) -> prior distribution what we know about parameters
  - P(data|parameters) -> likelihood the data according to statistical model

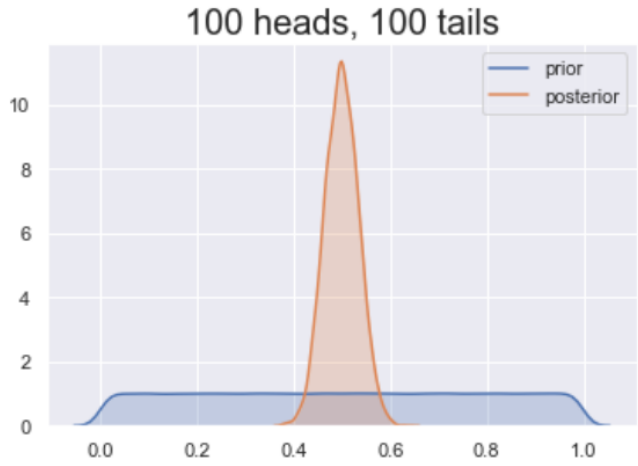
$$P(\text{parameters}|\text{data}) = \frac{P(\text{data}|\text{parameters}) * P(\text{parameters})}{P(\text{data})}$$

# Prior vs posterior distributions



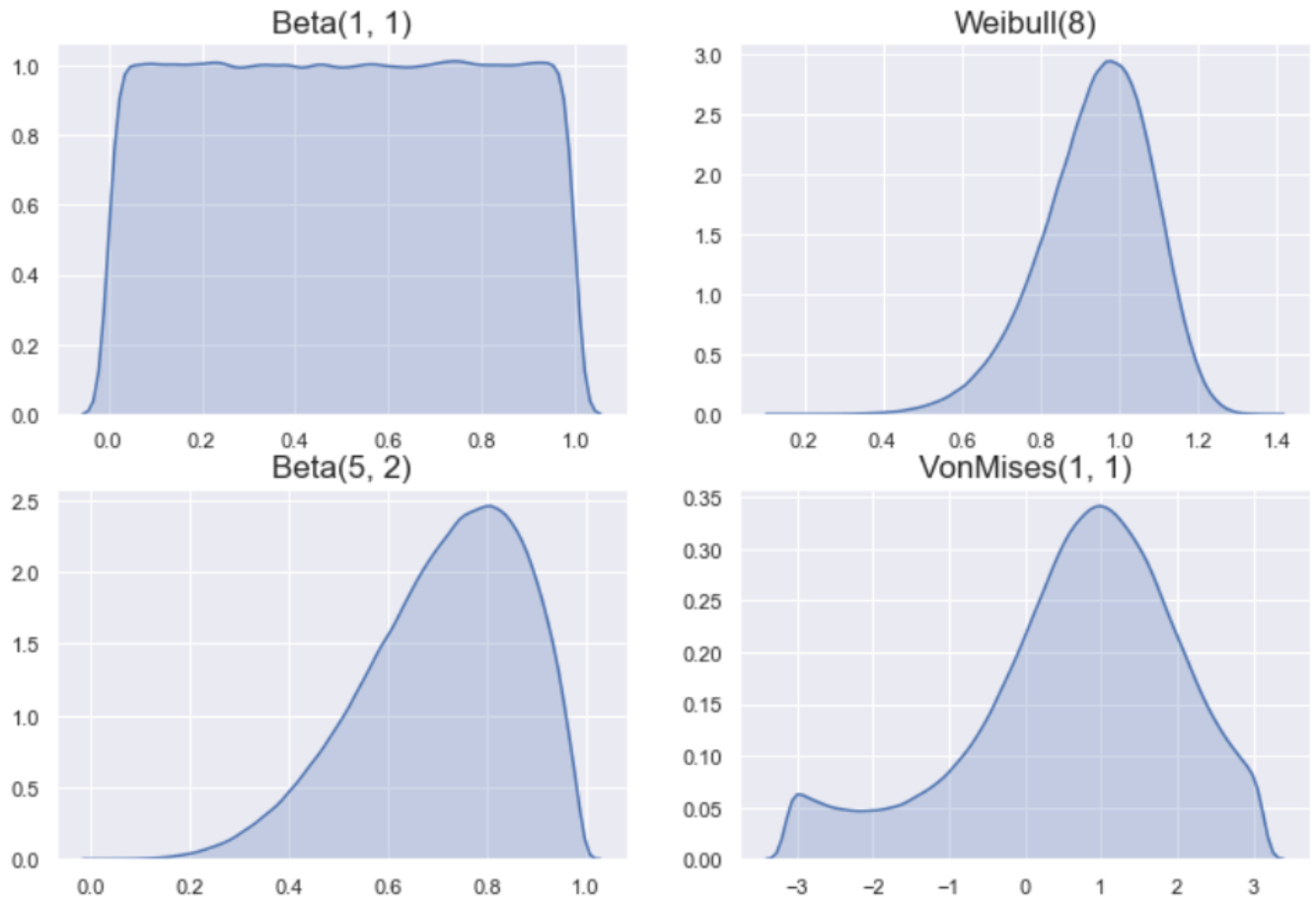



# Prior distribution

- Prior distribution what makes Bayesian inference possible with little data.
- You should explicitly justify your prior distribution choice, otherwise might be accused of cheating.
- It should be chosen before seeing the data.
- We assign alpha and beta values
- https://en.wikipedia.org/wiki/Conjugate_prior

Which one seems better prior distribution if you believe that the outcome ranges between %70 and %90, and rarely goes below than %50?



# Evaluating Bayesian results

- Instead of p-values, robustness of Bayesian results come from the distribution.
- Numeric equivalence of distributions are complicated, but still there are numeric expression.
- Credible intervals, instead of “confidence intervals”
  - They are not real intervals, are measures of uncertainty in the estimation.
  - In the Bayesian world, a parameter is a random variable, so we can talk about the probability of falling into some interval.
  - Frequentist approach can only make probabilistic statements about the interval, not the parameter.
- Highest Posterior Density, High Density Interval
  - Provides lower ad upper bounds
  - Pymc3 library in python

# Linear regression

- Frequentist models
  - β's are single numbers
  - Error term has a normal distribution with zero mean
- Bayesian models
  - We treat the response as a random variable, that has a normal distribution
  - We assign prior distributions for β's

# Linear regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$$

$$\text{sales} = \beta_0 + \beta_1 \text{marketingSpending}$$

- Frequentist inference:
  - $\text{sales} = \beta_0 + \beta_1 \text{marketingSpending} + \varepsilon$
  - $\varepsilon \sim \mathcal{N}(0, \sigma)$

- Bayesian inference:
  - $\text{sales} \sim \mathcal{N}(\beta_0 + \beta_1 \text{marketingSpending}, \sigma)$

$$\beta_0 \sim \mathcal{N}(5, 2)$$

$$\beta_1 \sim \mathcal{N}(2, 10)$$

$$\sigma \sim \mathcal{U}nif(0, 3)$$

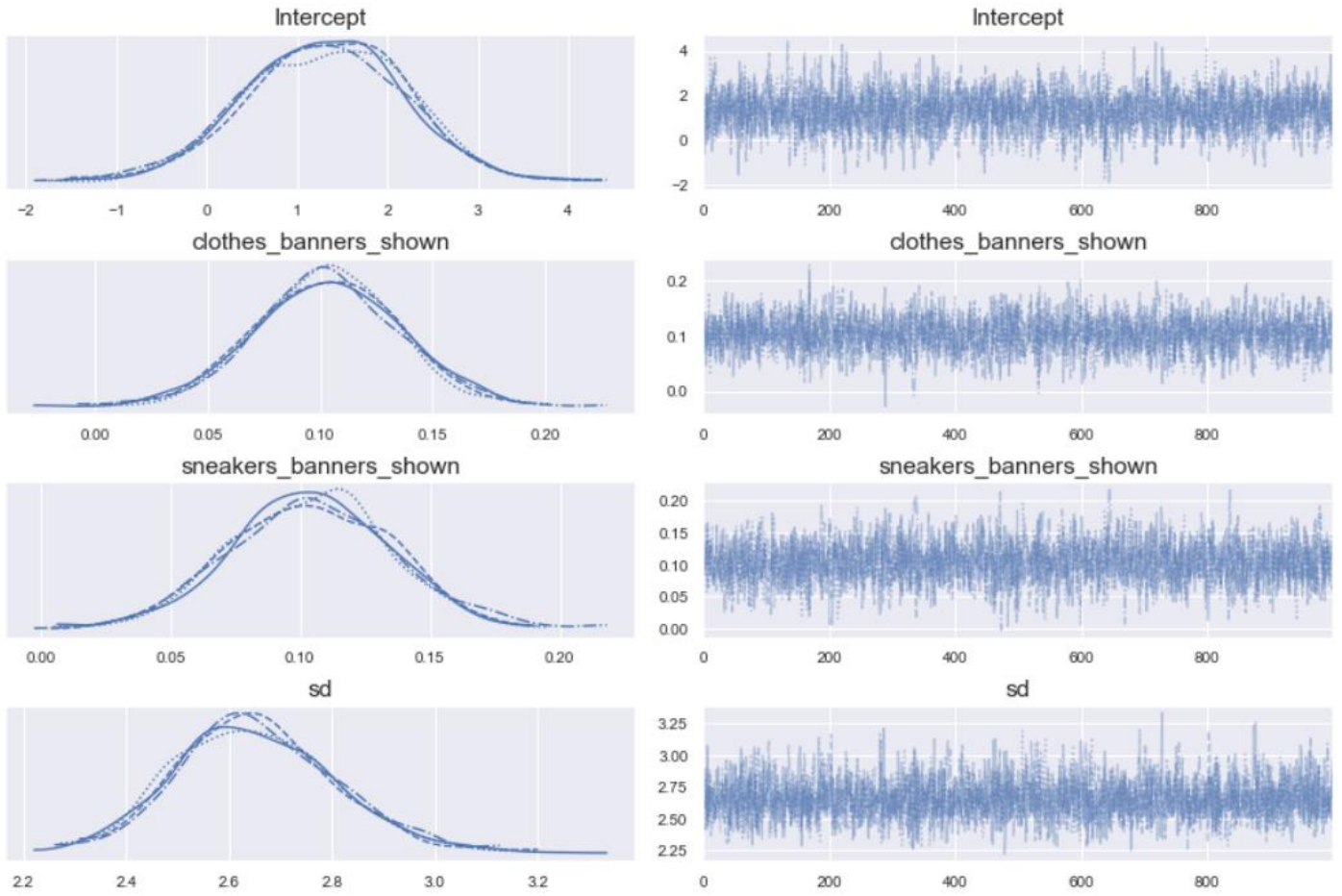
# Markov Chain Monte Carlo (MCMC)

- Sampling from known posterior requires conjugate priors
- Sampling from unknown posterior provides more flexibility
- Numbers generated by MCMC become draws from the posterior after many samples thanks to Markov Chain's convergence property.
- pymc3.model()
- pymc3.GLM.formula (formula, data = …)
  - Draws
  - Tune
  - Chain

```
formula = "num_clicks ~ clothes_banners + sneakers_banners"
with pm.Model() as model_1:
        pm.GLM.from_formula(formula, data=data_name)
        trace_1 = pm.sample(draws=1000, tune = 500)
```

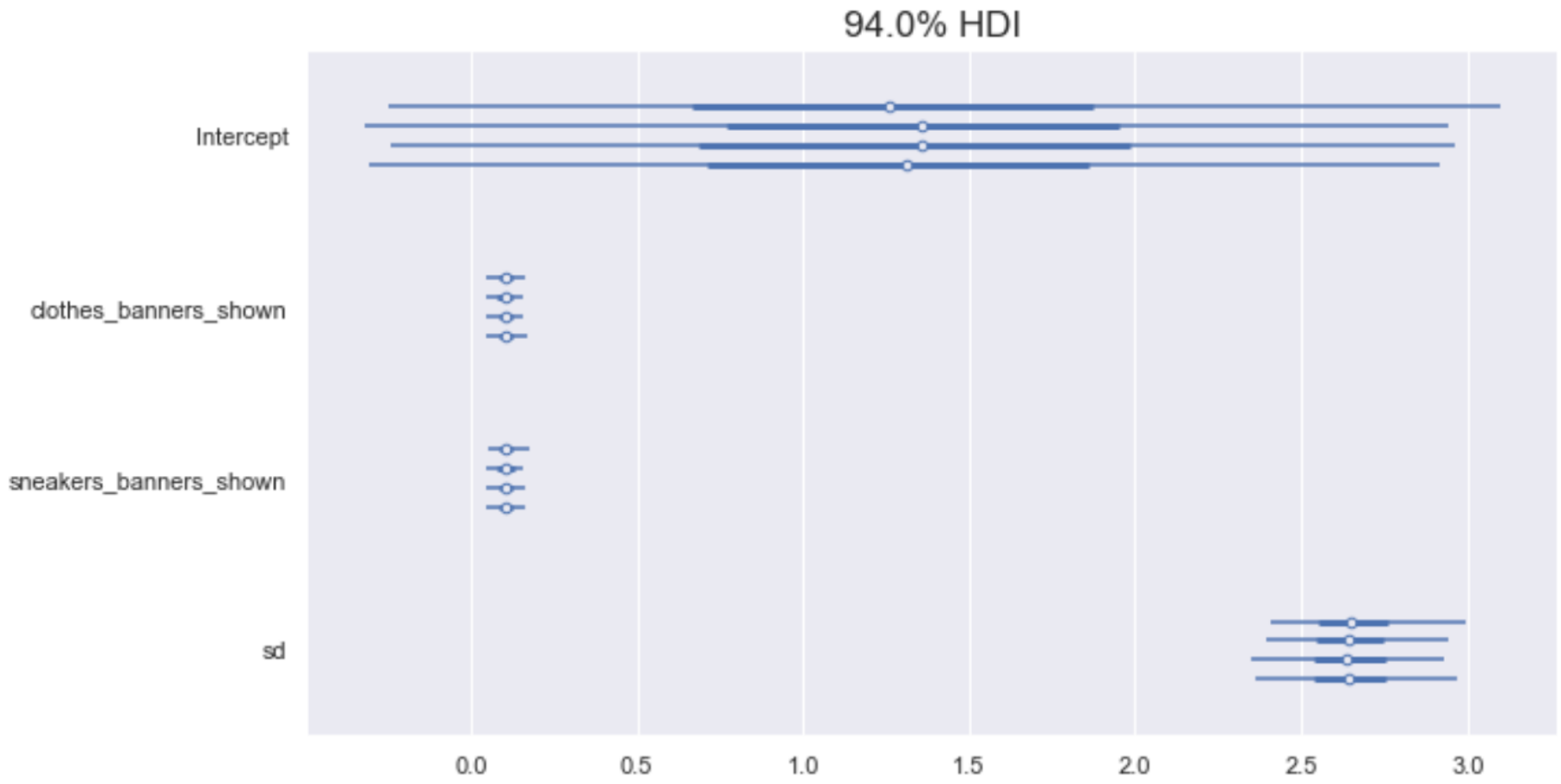
# Interpretation

pymc3.traceplot(trace_1)



# Interpretation

pymc3.forestplot(trace_1)



# Interpretation

- pycm3.summary(trace_1) shows parameter means, sd, hdi, etc.

- Test for chain convergence: R_hat should be equal to 1.0
  - Otherwise, some of your models did not convergence

- Compare models: The Widely Applicable Information Criterion, or WAIC.
  - The lower is better
  - pymc3.compare()
  - pymc3.compareplot()

# Making predictions

- We have a well-fitting model now, but then how can we test it with a new data?
- Sampling predicting draws and test error distribution

Therefore:

1. Fit a Bayesian model
2. Inspect the model to verify its correctness
3. Predict the outcome, to present errors

```
errors=[]
for index, test_example in ads_test.iterrows():
    error=posterior_predictive['y'][:,index] - test_example['num_clicks']
    errors.append(error)

error_distribution=np.array(errors).reshape(-1)
error_distribution.shape
```

# Challenges

- Mathematical derivation is more complex
- Higher computational cost
- Traditional predominance of frequentist approach

- Still growing exponentially!!!