

# 確率モデル入門

## 確率の用語整理、確率変数、確率分布

nepia271

Liberal Arts for Tech

2020/05/30

# 講義（セミナー前半）

## 目標

統計を勉強し始めた人が確率を理解しやすくなること

## アジェンダ

- ▶ 試行
- ▶ 標本空間と事象
- ▶ 確率変数
- ▶ 確率分布
  - ▶ 離散確率分布
  - ▶ 連続確率分布

# 全体の構成

- ▶ なぜ「確率」が必要なのか？
- ▶ 確率の用語整理
- ▶ 確率変数とは？
- ▶ 確率分布とは？
- ▶ Python で確率分布の例を見る
  - ▶ 離散確率分布
  - ▶ 連続確率分布

# なぜ「確率」が必要なのか？（１）

- ▶ 「記述統計」ならば「確率」は不要でした
  - ▶ ヒストグラムを書いてデータを概観する
  - ▶ 平均値、中央値を算出してデータの中心傾向を知る
  - ▶ 分散を算出してデータの散らばり具合を知る
  - ▶ ↑これらに確率の知識は不要

## なぜ「確率」が必要なのか？（２）

- ▶ 一方「統計モデル」を使うモチベーションは……
  - ▶ データの背後にある "原則" "真理" が知りたい  
例) データからわかる、歪んだコインを投げて表が出る%は？
  - ▶ しかも"定量的"に知りたい
    - ▶ (データから) どれくらい歪んでいるのか
    - ▶ (パラメータ推定が) どれくらい信頼できるのか  
(「サンプル数が多いほど信頼できる」とは言うけれど?)
- ▶ はたまた「統計モデリング」では……
  - ▶ 実世界のデータから筋のよい「確率モデル」を記述したい
  - ▶ そのモデルがどれくらい本当なのかを議論したい

# まずは確率の用語整理から……

※資料からかいつまんで紹介

## > 試行 (trial)

実験や観測などを行うことです。確率モデルに基づいて論理展開を行っていくにあたっては、試行を行った結果を確率的に解釈していきます。

## > 標本空間 (sample space)

試行の結果を要素とする集合です。

## > 事象 (event)

標本空間の部分集合です。

# 確率変数とは？

確率変数とは「ランダムな値をとる」変数です

- ▶ (注意) 確率変数は〇〇ではありません
  - ▶ 確率変数は「とる値の集合」ではありません  
それは「標本空間」です
  - ▶ 確率変数は「具体的な実現値」ではありません  
それは「試行」です
  - ▶ 確率変数は「起こりうる事柄」ではありません  
それは「事象」です

(いやあ、難しいですね……)

## 余談：ランダムという概念は実はとっても難しい

- ▶ 20 世紀初頭のこと
- ▶ アンドレイ・コルモゴロフというロシアの数学者がいて
- ▶ 集合論・測度論・ルベーグ積分を駆使して確率を定式化した  
(公理的確率論)
- ▶ それまではランダム性について  
上手に表現することができていなかった  
(厳密さを欠いていた)

ランダムも、確率変数も、  
むずかしいので、  
深く考えないことがオススメです。



# 確率分布とは？

- ▶ 「各々の値をとる確率」を表す分布
  - ▶ 例： コインを投げたとき……
  - ▶ コインが表を取る確率： 50%
  - ▶ コインが裏を取る確率： 50%
- ▶ ヒストグラムをサンプル数で割ったものと「似ています」
  - ▶ じつのところ「そうではない」のですが
  - ▶ 無限にデータ数があるなら一致していきます

データがまずあって、  
確率分布へのあてはまりを考える流れが多いです。

# 離散確率分布・連続確率分布

- ▶ 離散確率分布とは？
  - ▶ 確率変数  $X$  が離散値をとる場合の確率分布です
  - ▶ 例) コインの表裏、サイコロの出目
  - ▶ 一様分布
  - ▶ 二項分布
- ▶ 連続確率分布とは？
  - ▶ 確率変数  $X$  が連続値をとる場合の確率分布です
  - ▶ 例) 花卉の長さ、16 歳男子の身長
  - ▶ 正規分布
  - ▶ ポアソン分布

離散ならとりうる値ごとに確率が出せますが、  
連続ではそうはいきません。

「身長が 170cm ジャスト」とはいえないからです。

考えてもよくわからない……

- ▶ Python で動かしてみよう
- ▶ 実際に動いている様子で理解のヒントになるかも

## 離散確率分布の例：サイコロ

## 連続確率分布の例：正規分布

サンプルを増やすと真の分布に漸近する

# 講義のまとめ

- ▶ 確率モデリングに入門するにはまず確率
- ▶ 確率の基礎事項は混同しやすい
- ▶ 確率を一発で理解しようとするのは大変

ではどうすれば？

- ▶ ゆっくり丁寧にやることで理解する
- ▶ Python で動かすことで理解する
  - ▶ 紙とペンでやるよりも
  - ▶ Python のほうがカンタンです

少し休憩をしたのち、ハンズオンに入ります。

# ハンズオン

- ▶ `sklearn.dataset` を読み込む
- ▶ 適当なヒストグラムを書く
- ▶ 何の数学的分布に近いか見てみる

その他、素朴な疑問について  
確率に限らず拾っていく時間とします