情報科学研究２
データ解析特論
第2回　確率分布と統計モデル

情報科学領域
平居　悠（ひらい　ゆたか）

到達目標

- 基本的な統計モデルについて説明できる。
- 観測データに適切な統計モデルを適用し、解析できる。
- 定量的なデータ解析結果を示すことができる。

前回

第1回　データの理解
第2回　確率分布と統計モデル
第3回　確率分布と統計モデルの最尤推定
第4回　一般化線形モデル
第5回　統計モデルの選択
第6回　尤度比検定と検定の非対称性
第7回　一般化線形モデルの応用
第8回　一般化線形混合モデル
第9回　マルコフ連鎖モンテカルロ法
第10回　ベイズ統計モデル
第11回　一般化線形モデルのベイズモデル化
第12回　事後分布の推定
第13回　階層ベイズモデル
第14回　空間構造のある階層ベイズモデル
第15回　まとめ

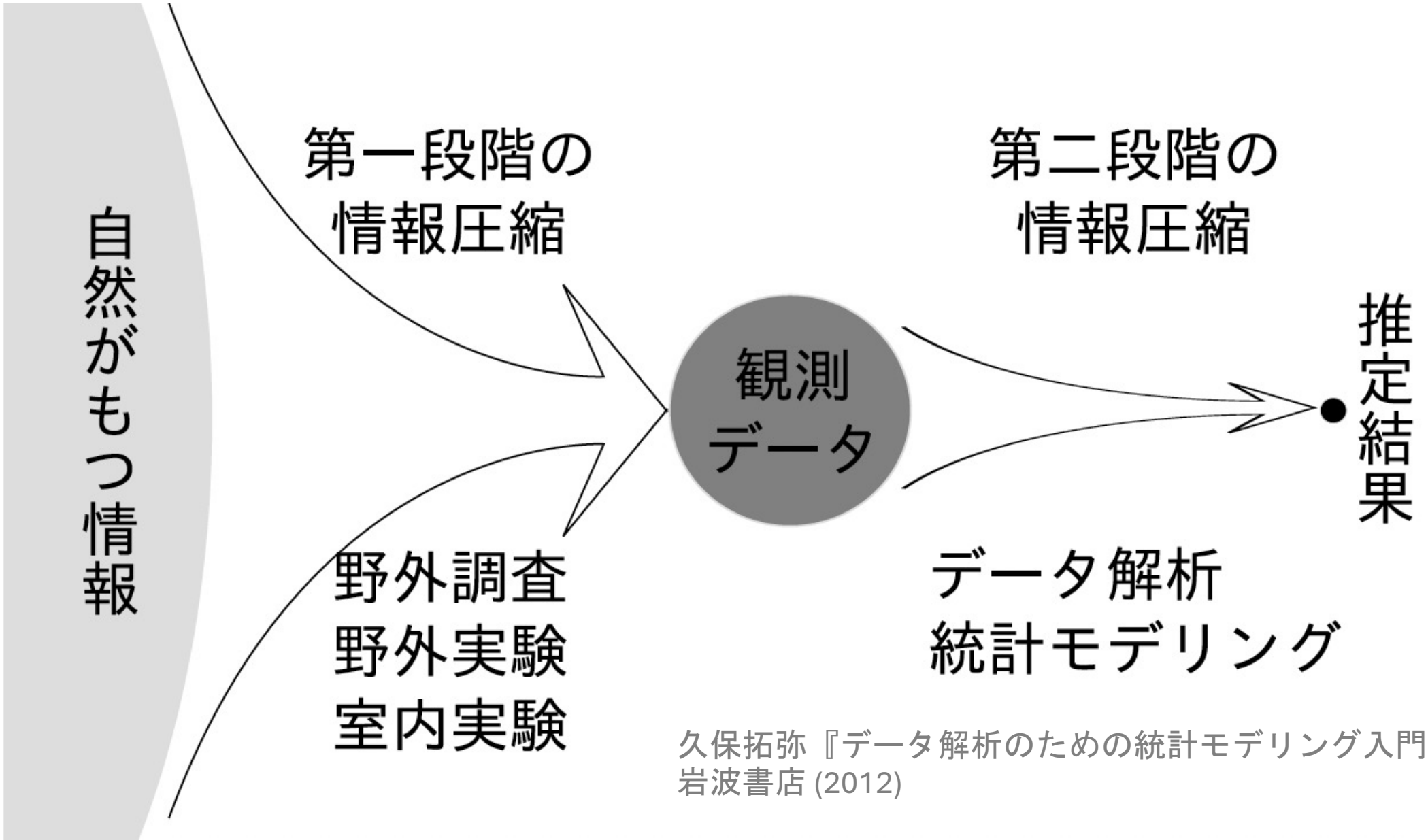
前回の目標

統計モデルとは何か説明できる。

前回の内容

統計解析の必要性
統計モデル
プログラミング言語Python

情報消失・情報圧縮



久保拓弥『データ解析のための統計モデリング入門』
岩波書店 (2012)

統計解析の必要性

データ化した情報は統計的な方法で扱うしかない。

統計解析は客観的で再現性がある。

データ解析とは

収集されたデータを分析した結果から、一定の法則性や共通点を見出し、問題の解決に役立てること

統計モデル

観測データやそのデータに見られる様々なパターンをうまく説明できる確率論的な数理モデル

統計モデルの特徴

統計モデルの部品は確率分布

- 確率分布の形はパラメーターによって決まる。

なぜPythonを学ぶのか？

データ解析や人工知能が得意

充実したライブラリ（再利用可能なコードの集合体）

データ分析：Numpy, Pandas, Matplotlibなど
機械学習：Scikit-learn, TensorFlowなど
Web開発：Flask, Djangoなど

課題

Pythonのインストールを完了させ、授業資料p.61–66のプログラムを実行する。

提出方法：第2回授業冒頭で実行結果を確認

今回

第1回　データの理解
第2回　確率分布と統計モデル
第3回　確率分布と統計モデルの最尤推定
第4回　一般化線形モデル
第5回　統計モデルの選択
第6回　尤度比検定と検定の非対称性
第7回　一般化線形モデルの応用
第8回　一般化線形混合モデル
第9回　マルコフ連鎖モンテカルロ法
第10回　ベイズ統計モデル
第11回　一般化線形モデルのベイズモデル化
第12回　事後分布の推定
第13回　階層ベイズモデル
第14回　空間構造のある階層ベイズモデル
第15回　まとめ

今回の目標

データと確率分布の対応をプロットできる。

今回の内容

種子数の統計モデリング

データと確率分布の対応関係

ポアソン分布

今回の内容

種子数の統計モデリング

データと確率分布の対応関係

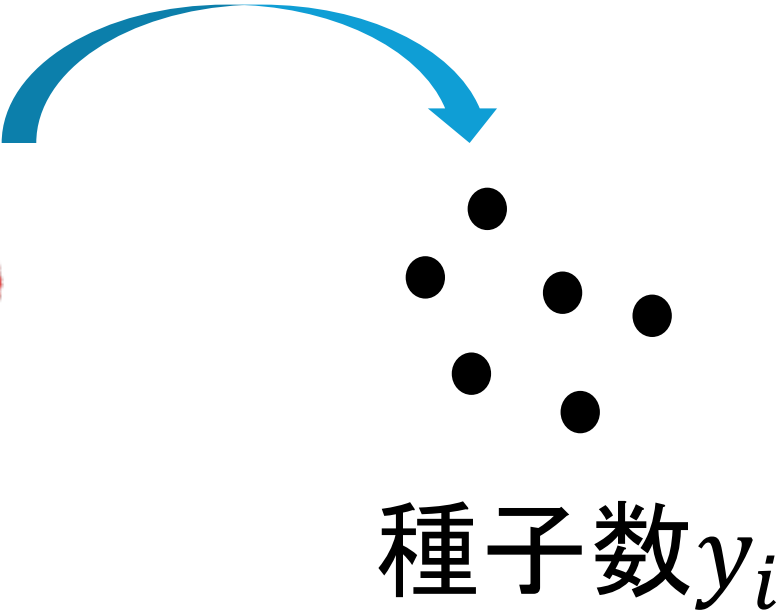
ポアソン分布

例題：植物種子数の統計モデリング

調査対象：架空の植物50個体

データ：各個体の種子数

個体i



種子数y_i

データ保存用ディレクトリの作成

/projects/data_analysis内に“data”ディレクトリを作成

データのダウンロード

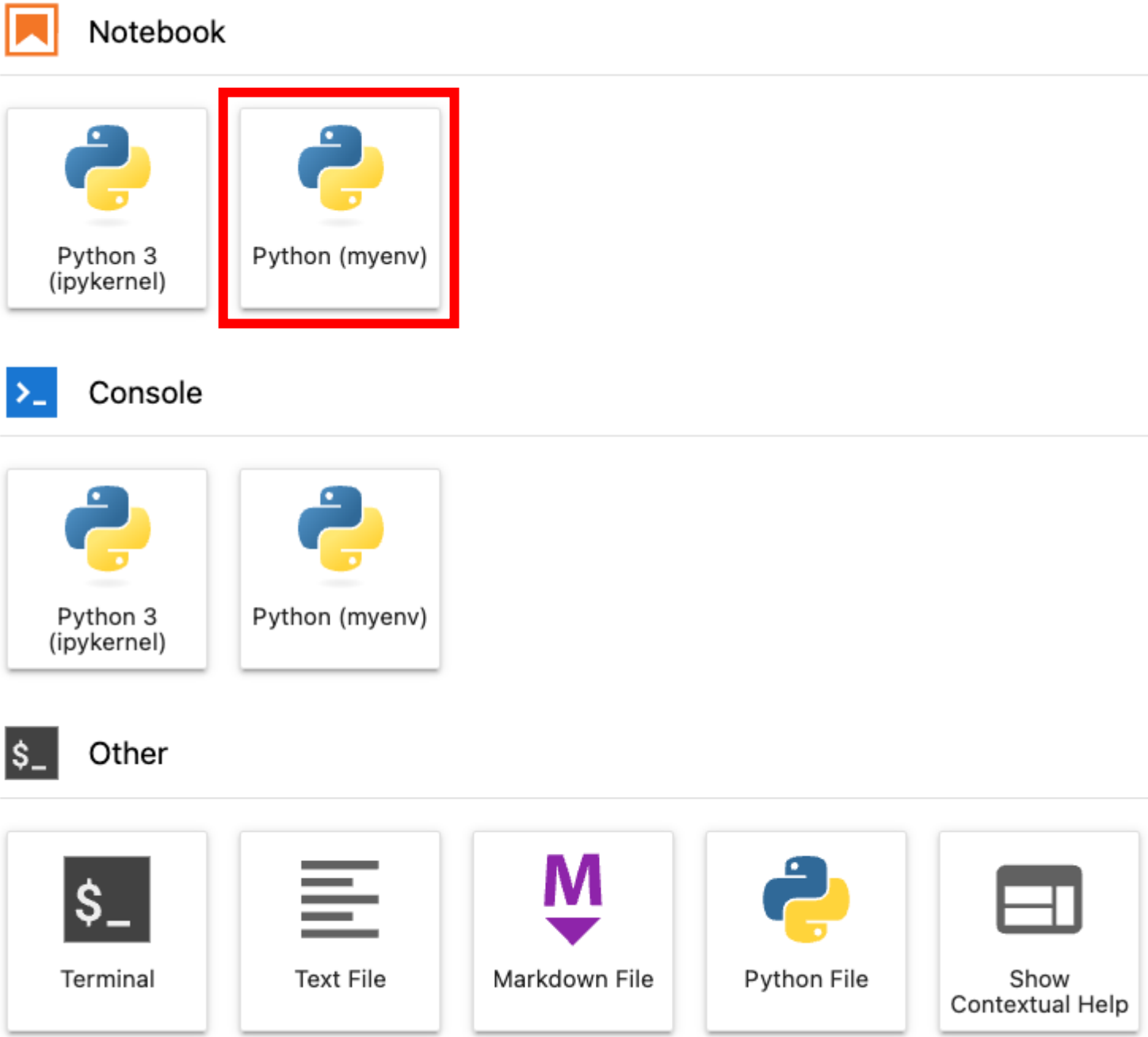
下記URLからデータをダウンロードし、dataディレクトリへ保存

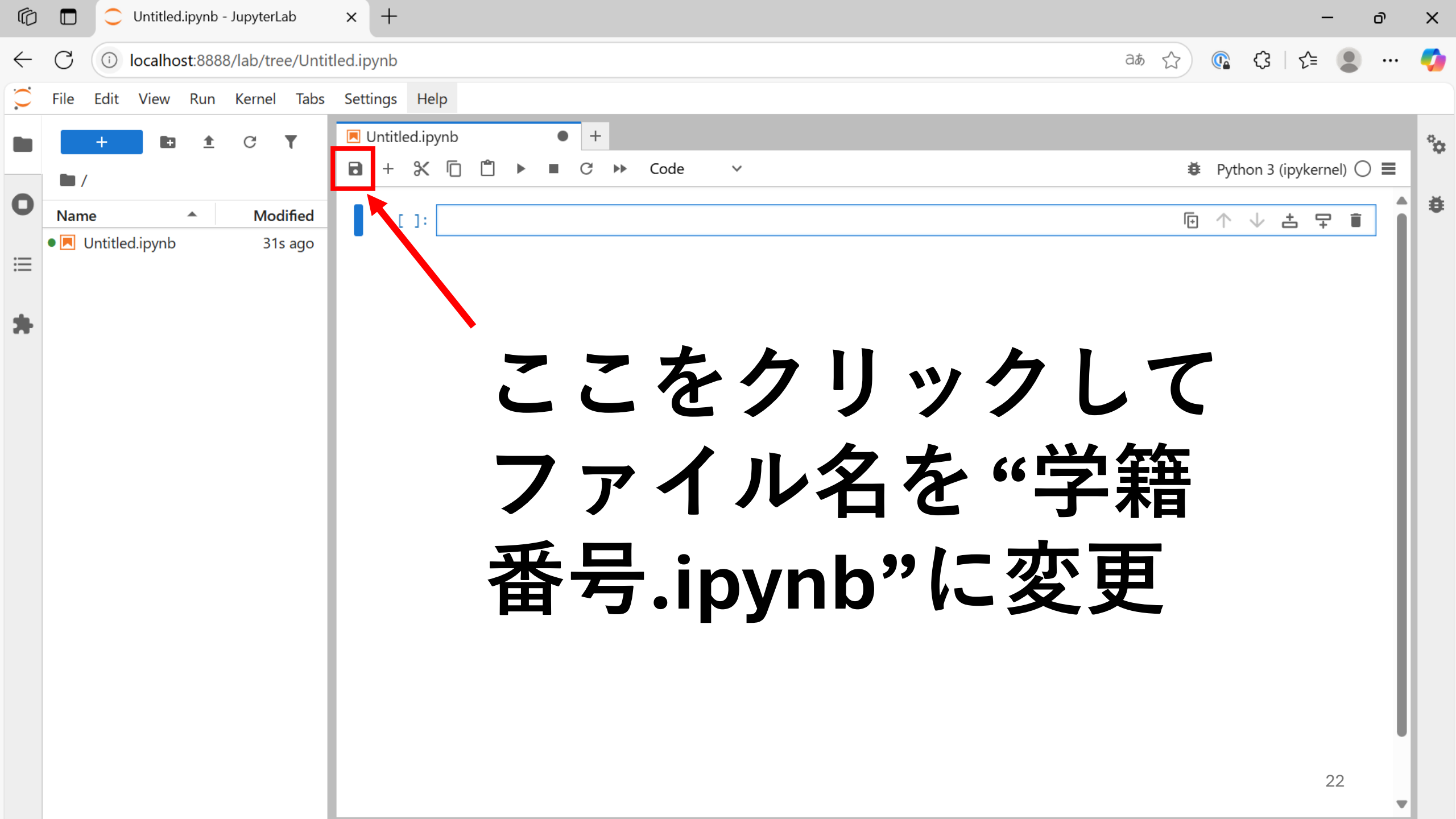
https://kuboweb.github.io/-kubo/stat/iwanamibook/fig/distribution/data.RData

JupyterLab起動

/projects/data_analysis/に移動し、JupyterLabを起動する。

新しいノートブックの生成




Untitled.ipynb - JupyterLab
localhost:8888/lab/tree/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help
/
Name
Modified
Untitled.ipynb
31s ago
Untitled.ipynb
Code
Python 3 (ipykernel)
[ ]:
ここをクリックして
ファイル名を “学籍
番号.ipynb”に変更

ライブラリのインポート

Shift+Enterで実行

```
### インポート

# 数値計算
import math
import numpy as np
import pandas as pd

# 統計計算
import scipy.stats as stats

# Rデータセットの読み込み
import rdata

# 可視化
import matplotlib.pyplot as plt
plt.rcParams['font.family'] = 'Meiryo'
```

データの読み込みと表示

```
# RdataをPythonオブジェクトに変換
converted = rdata.read_rda('./data/ch02/data.RData', default_encoding='ASCII')
converted
```

pandasデータフレーム化

```
data = pd.DataFrame(converted['data'], columns=['種子数']).astype(int)
```

結果の表示

```
print('data.shape: ', data.shape)
display(data.head())
```

dataの内容表示

```
data['種子数'].values
```

dataに含まれるデータ数の確認

```
len(data)
```

dataの要約統計量の表示

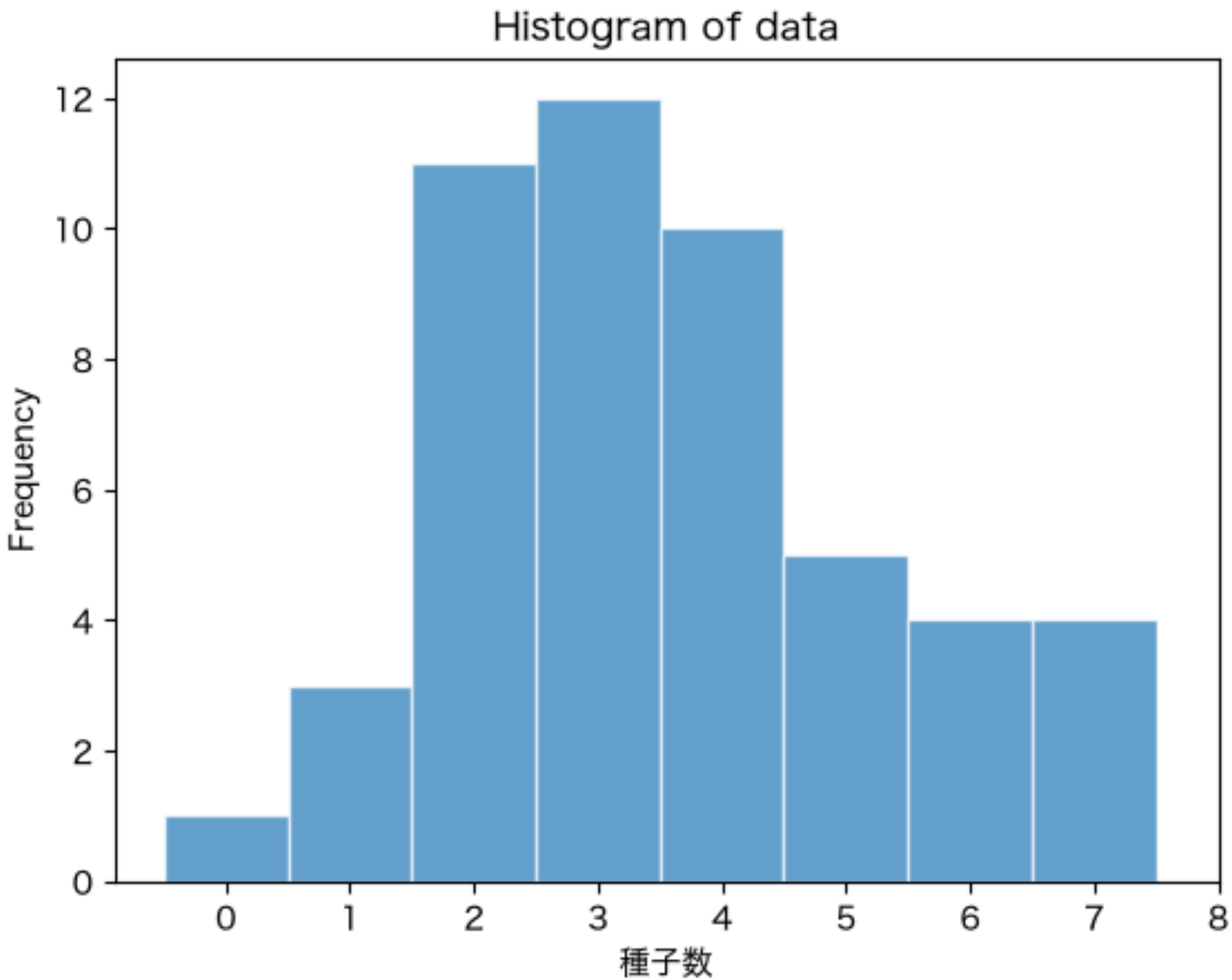
```
data.describe().T
```

度数分布を表示

```
data.value_counts(sort=False).to_frame().T
```

ヒストグラムの描画

```
data.plot.hist(
    bins=np.arange(-0.5, 8), xticks=range(9), edgecolor='white', alpha=0.7,
    legend=False, title='Histogram of data', xlabel='種子数'
);
```


Histogram of data
Frequency
種子数
0
2
4
6
8
10
12
0
1
2
3
4
5
6
7
8

標本分散の表示　データのばらつき

```
data['種子数'].var()
```

標本標準偏差の表示

```
data['種子数’].std()
```

今回の内容

種子数の統計モデリング

データと確率分布の対応関係

ポアソン分布

種子数データの特徴

- 個数を数えられるカウントデータ
- 1個体の種子数の標本平均は3.56
- 個体ごとに種子数に**ばらつき**があり、ヒストグラムを描くとひと山の分布になる。

ばらつきの表現→確率分布

確率分布

- 確率変数の値とそれが出現する確率を対応させたもの。
  - 例：ある植物個体iの種子数y_i

- パラメータの値に依存して分布の形が変わる。

平均3.56のポアソン分布の確率分布の計算

```
# 確率変数yの値の設定
y = np.arange(10)

# yについて平均3.56のポアソン分布の確率質量関数を算出 scipy.stats利用
prob = stats.poisson.pmf(k=y, mu=data.mean())
prob
```

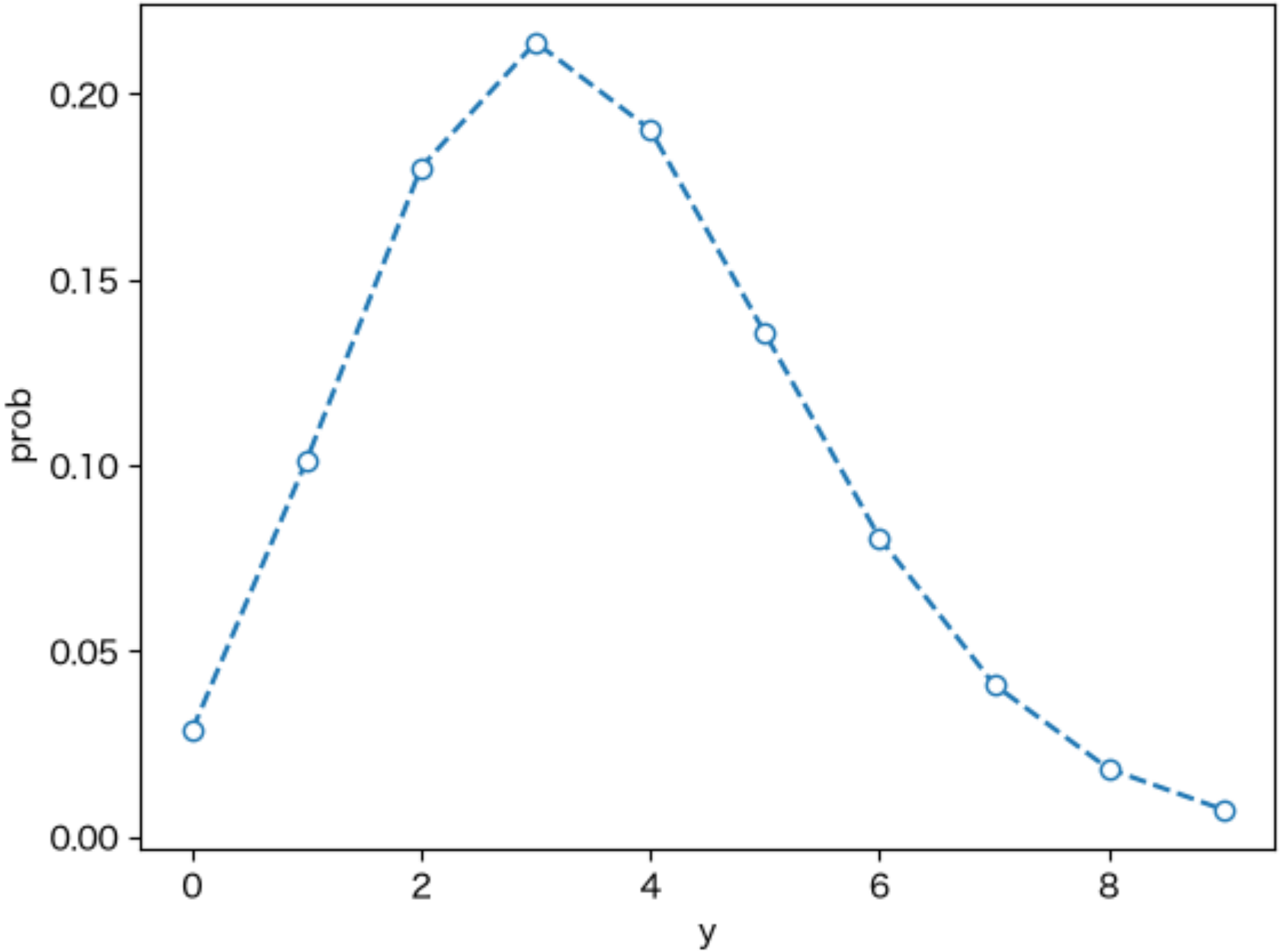
データフレーム化

```
prob_df = pd.DataFrame({'y': y, 'prob': prob})
prob_df
```

	y	prob
0	0	0.028439
1	1	0.101242
2	2	0.180211
3	3	0.213851
4	4	0.190327
5	5	0.135513
6	6	0.080404
7	7	0.040891
8	8	0.018197
9	9	0.007198

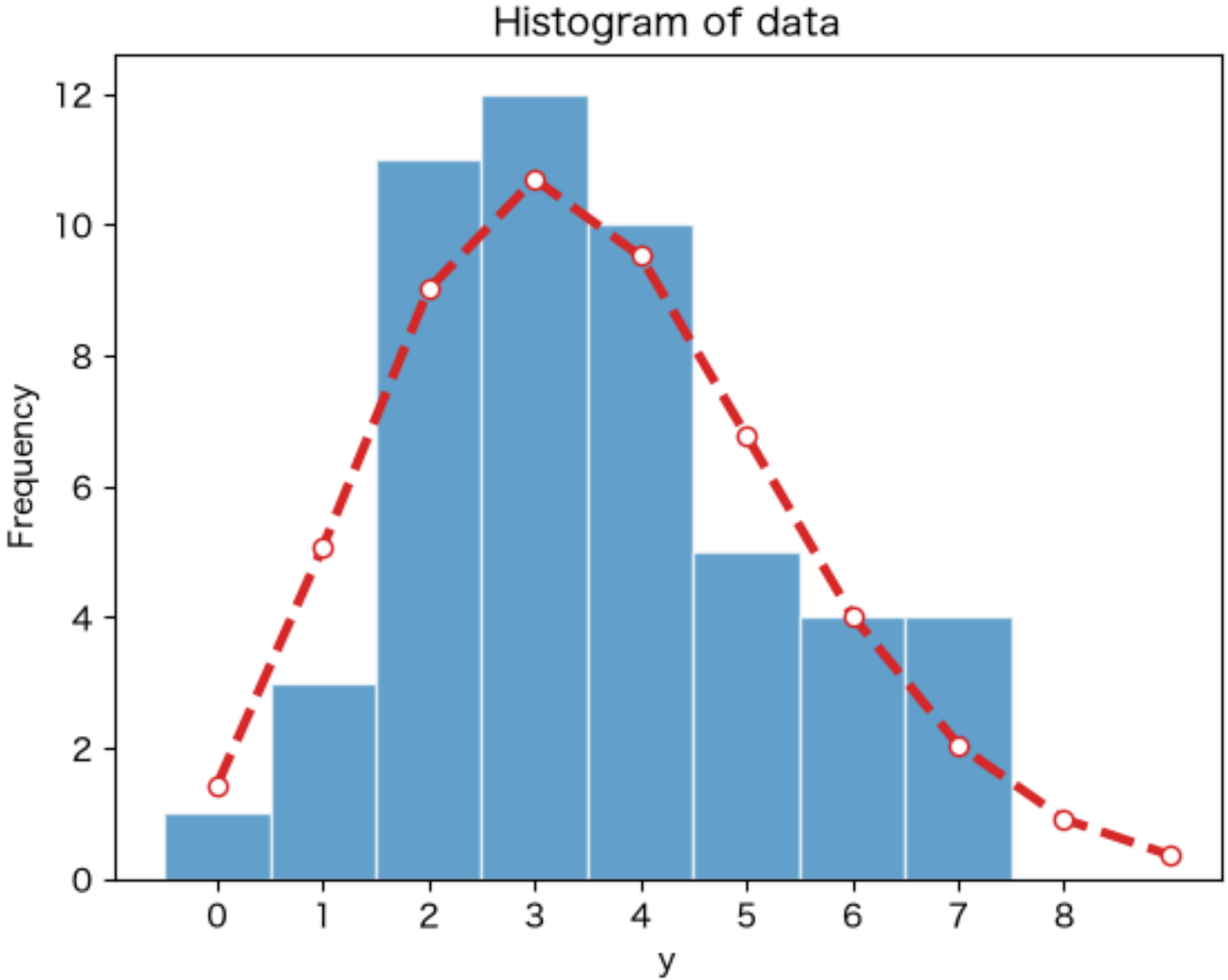
種子数yとその確率の関係の描画

```
prob_df.plot(x='y', y='prob', marker='o', mfc='white', ls='--',
             ylabel='prob', legend=False);
```


0.20
0.15
0.10
0.05
0.00
prob
0
2
4
6
8
y

観測データと確率分布の対応の描画

```
# ヒストグラムの描画
ax = data.plot.hist(bins=np.arange(-0.5, 8), xticks=range(9),
                    color='tab:blue', edgecolor='white', alpha=0.7,
                    legend=False, title='Histogram of data')
# 個体数の予測値の折れ線グラフの描画
prob_df['pred'] = prob_df['prob'] * len(data)
prob_df.plot(x='y', y='pred', marker='o', color='tab:red', mfc='white',
             lw=3, ls='--', legend=False, ax=ax);
```


Histogram of data
Frequency
y
0
2
4
6
8
10
12
0
1
2
3
4
5
6
7
8

今回の内容

種子数の統計モデリング

データと確率分布の対応関係

ポアソン分布

ポアソン分布

確率変数をy、平均パラメータをλとするとポアソン分布の確率質量関数は、

$$p(y|\lambda) = \frac{\lambda^y \exp(-\lambda)}{y!},$$

となる。

ポアソン分布の性質

- $y \in \{0,1,2,\cdots,\infty\}$の値をとり、すべての$y$について和をとると1になる。

$$\sum_{y=0}^{\infty} p\,(y|\lambda) = 1,$$

- 確率分布の平均はλである。

$$\lambda \geq 0,$$

- 分散と平均は等しい。

λ= 平均 = 分散.

ポアソン分布の確率質量関数を算出する関数の定義

```
def poisson_pmf(y, lam):
    return lam**y * np.exp(-lam) / np.array([math.factorial(i) for i in y])
```

データと平均パラメータの設定

```
y = np.arange(10)
lam = 3.5
```

確率の算出

```
poisson_pmf(y=y, lam=lam)
```

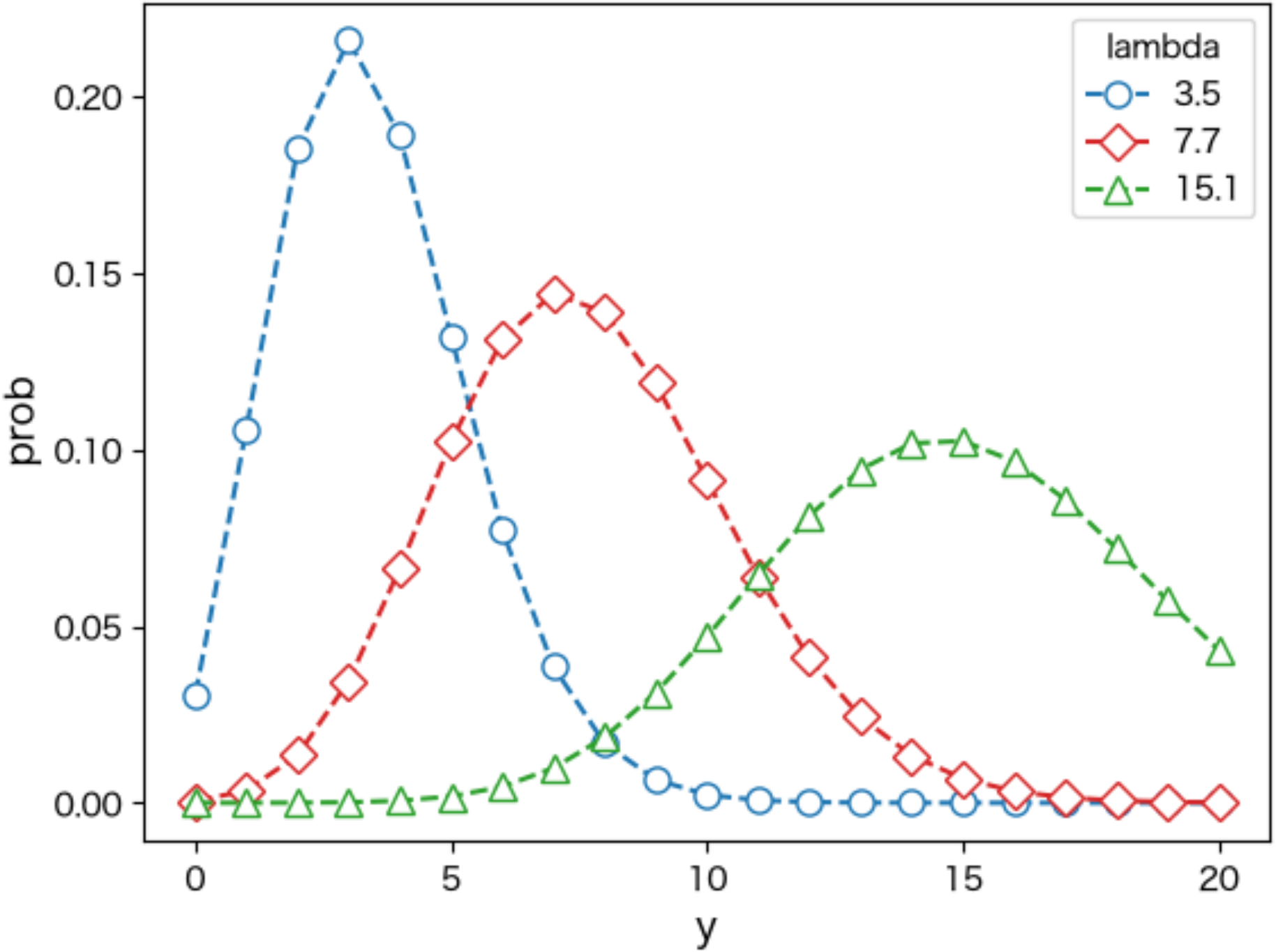
scipy.statsを用いた確率の算出

```
stats.poisson.pmf(k=y, mu=lam)
```

様々な平均のポアソン分布の描画

```
## 設定と準備
lams = [3.5, 7.7, 15.1]                  # 平均λ
markers = ['o', 'D', '^']                # マーカーの形状
colors = ['tab:blue', 'tab:red', 'tab:green'] # グラフの色
y = np.arange(21)                        # x軸の値（確率変数yの値）

## 描画
# ３つの平均ごとに確率算出と折れ線グラフ描画を繰り返し処理
for lam, marker, color in zip(lams, markers, colors):
    # 平均λを用いて確率を算出
    prob = stats.poisson.pmf(k=y, mu=lam)
    # 折れ線グラフの描画
    plt.plot(y, prob, c=color, mfc='white', marker=marker, ms=8, ls='--',
             label=f'{lam}')
# 修飾：x軸ラベル、y軸ラベル、x軸目盛り、凡例
plt.xlabel('y', fontsize=12)
plt.ylabel('prob', fontsize=12)
plt.xticks(ticks=range(0, 21, 5))
plt.legend(title='lambda');
```


lambda
3.5
7.7
15.1
prob
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
y

今回の内容

種子数の統計モデリング

データと確率分布の対応関係

ポアソン分布

今回の目標

データと確率分布の対応をプロットできる。

課題

本日のプログラミング演習で終わらなかった部分を完成させる。

提出方法：次回の授業時に実行結果を確認

次回

第1回　データの理解
第2回　確率分布と統計モデル
第3回　確率分布と統計モデルの最尤推定
第4回　一般化線形モデル
第5回　統計モデルの選択
第6回　尤度比検定と検定の非対称性
第7回　一般化線形モデルの応用
第8回　一般化線形混合モデル
第9回　マルコフ連鎖モンテカルロ法
第10回　ベイズ統計モデル
第11回　一般化線形モデルのベイズモデル化
第12回　事後分布の推定
第13回　階層ベイズモデル
第14回　空間構造のある階層ベイズモデル
第15回　まとめ