pythonで体験するベイズ推論 2.2.5-2.4

石田研 B4 池田 光

二項分布

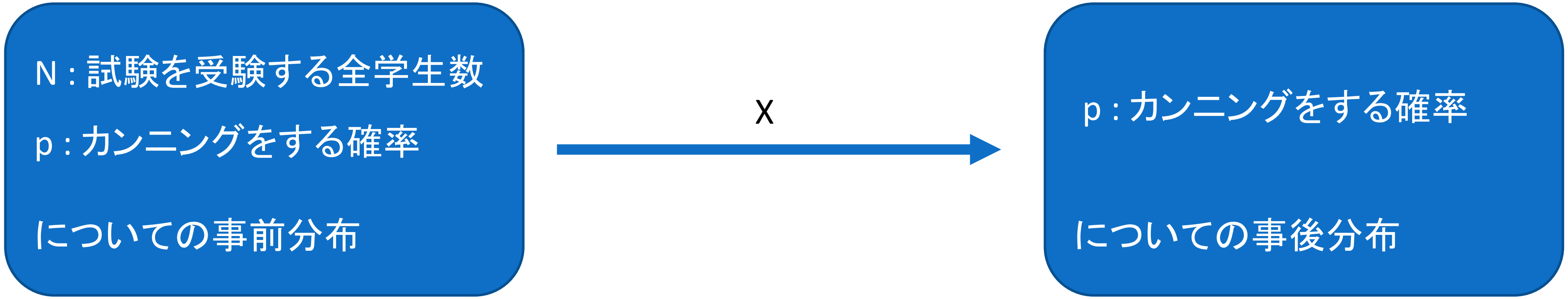
ベルヌーイ試行：何かを行ったときに起こる結果が2つしかない試行（ex コイントス）

⇛ベルヌーイ試行（成功確率pとする）をn回行って、成功する回数Xが従う確率分布を「二項分布」という

$$P(X=k)=\binom{N}{k}p^k(1-p)^{N-k}$$

例題：カンニングをした学生の割合

学生が試験中にカンニングをする頻度を求める。

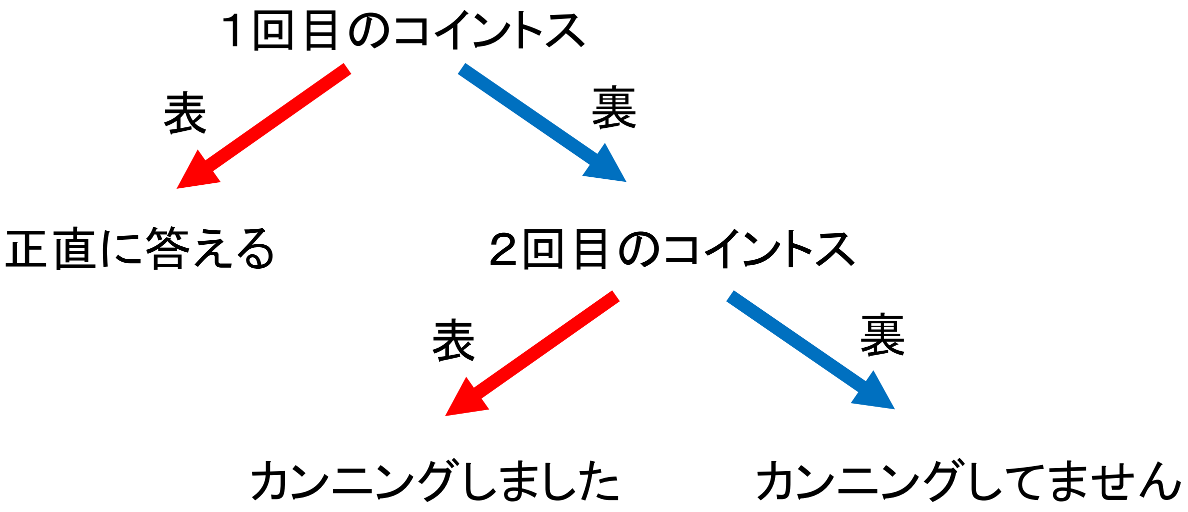


X : 試験後に結果を伝えず面接をし、カンニングしたという回答を得る回数

嘘をつくかも. . . .

プライバシーアルゴリズム

・プライバシーを保ちつつ、個々の学生に正直に答えるように促すアルゴリズム



⇛ これにより、嘘の回答がある可能性を排除

方法1 データ生成

・カンニングをした真の割合であるpを事前分布からサンプリング

```
import pymc as pm
N = 100
p = pm.Uniform("freq_cheating", 0, 1) #カンニングの割合
```

・ベルヌーイ分布にしたがう確率変数を割り当てる(0: カンニングしてない　1:カンニングした)

```
true_answers = pm.Bernoulli("truths", p, size=N) #真実
```

・1回目と2回目のコイントスをモデリング

```
first_coin_flips = pm.Bernoulli("first_flips", 0.5, size=N)
second_coin_flips = pm.Bernoulli("second_flips", 0.5, size=N)
```

方法1「はい」という割合の観測値

```
@pm.deterministic
def observed_proportion(t_a=true_answers, fc=first_coin_flips, sc=second_coin_flips):
  observed = fc * t_a + (1-fc)*sc
  return observed.sum()/float(N)
```

(1) 1回めのコイントスで表 & その学生がカンニングした
(2) 1のコイントスで裏 & 2回目のコイントスで表
の場合に限り、1になる

- 観測データについて

```
print(observed_proportion.value)
```

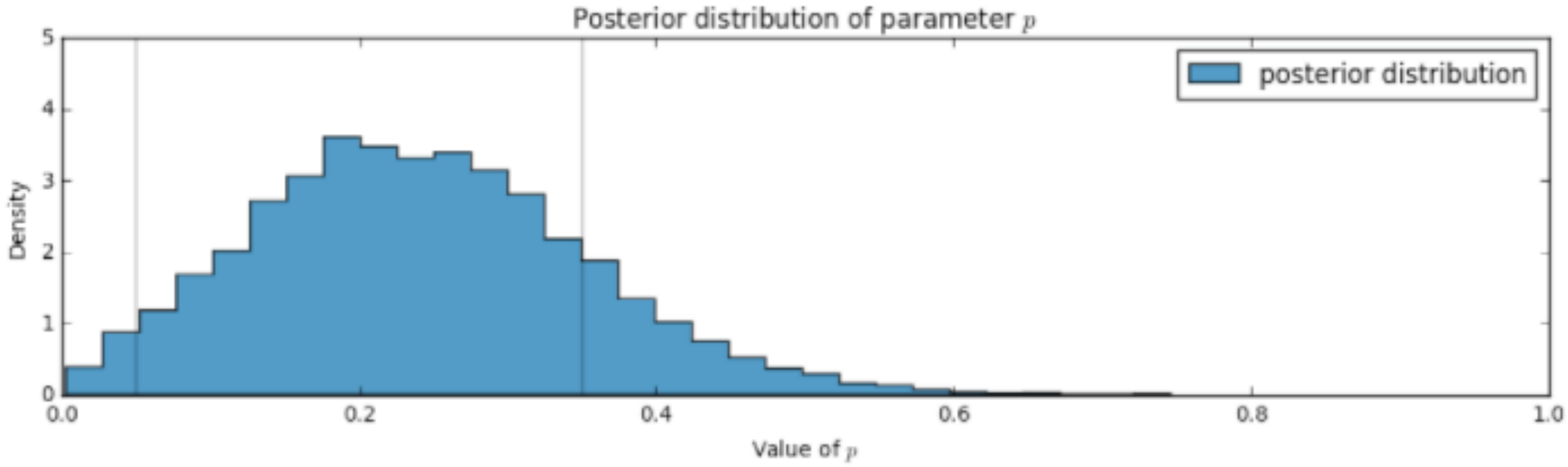
カンニングがまったくなかった場合、「はい」は1/4の確率
全員がカンニングをした場合、「はい」は3/4の確率

以降は、「はい」という回答が35だとする

方法1 モデルに適応

```
X = 35
observations = pm.Binomial("obs", N, observed_proportion, observed=True, value=X)
```

```
model = pm.Model([p, true_answers, first_coin_flips, second_coin_flips, observed_proportion, observations])

#以下は3章で
mcmc = pm.MCMC(model)
mcmc.sample(40000, 15000)
```



方法2

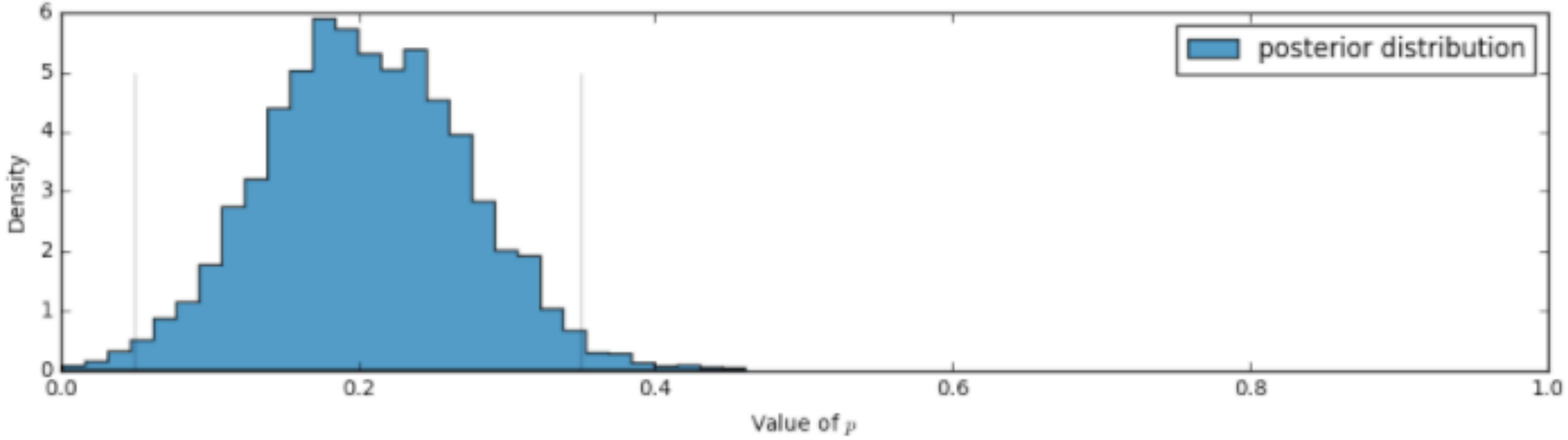
- 仮にpの値が与えられたら、学生が「はい」と回答する確率を求める事ができる

$$
\begin{aligned}
P(「はい」) &= P(\text{最初のコイン投げが表})P(\text{カンニングをした}) \\
&\quad + P(\text{最初のコイン投げが裏})P(\text{2 回目のコイン投げが表}) \\
&= \frac{1}{2}p + \frac{1}{2}\cdot\frac{1}{2} \\
&= \frac{p}{2} + \frac{1}{4}
\end{aligned}
$$

```
p = pm.Uniform("freq_cheating", 0, 1) #カンニングの割合

@pm.deterministic
def p_skewed(p=p):
    return 0.5 * p + 0.25
```

方法2 モデルに適応

```
yes_responses = pm.Binomial("number_cheaters", 100, p_skewed, value=35, observed=True)

model = pm.Model([yes_responses, p_skewed, p])

#以下は3章で
mcmc = pm.MCMC(model)
mcmc.sample(25000, 2500)
```

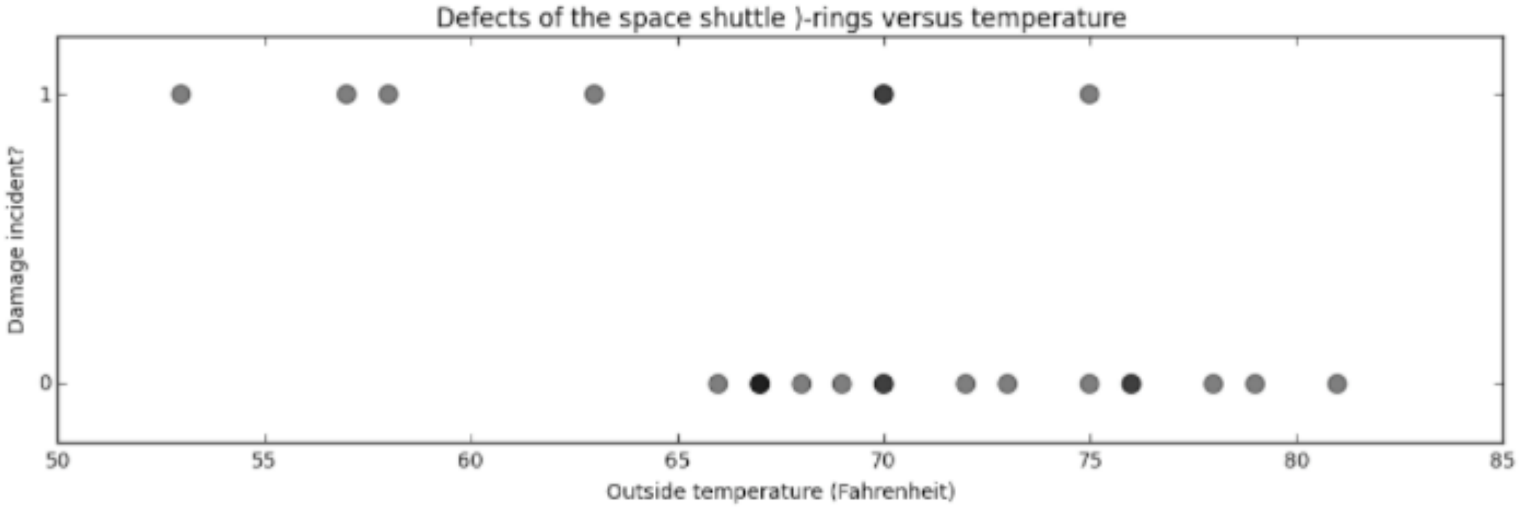


例題：「チャレンジャー号」の悲劇

1986年1月28日にアメリカで起きたスペースシャトル「チャレンジャー号」の墜落事故

事故はロケットブースターのOリングが原因

Oリングは外気温に敏感で、過去24回のフライトに対して不良についてのデータが23回あった



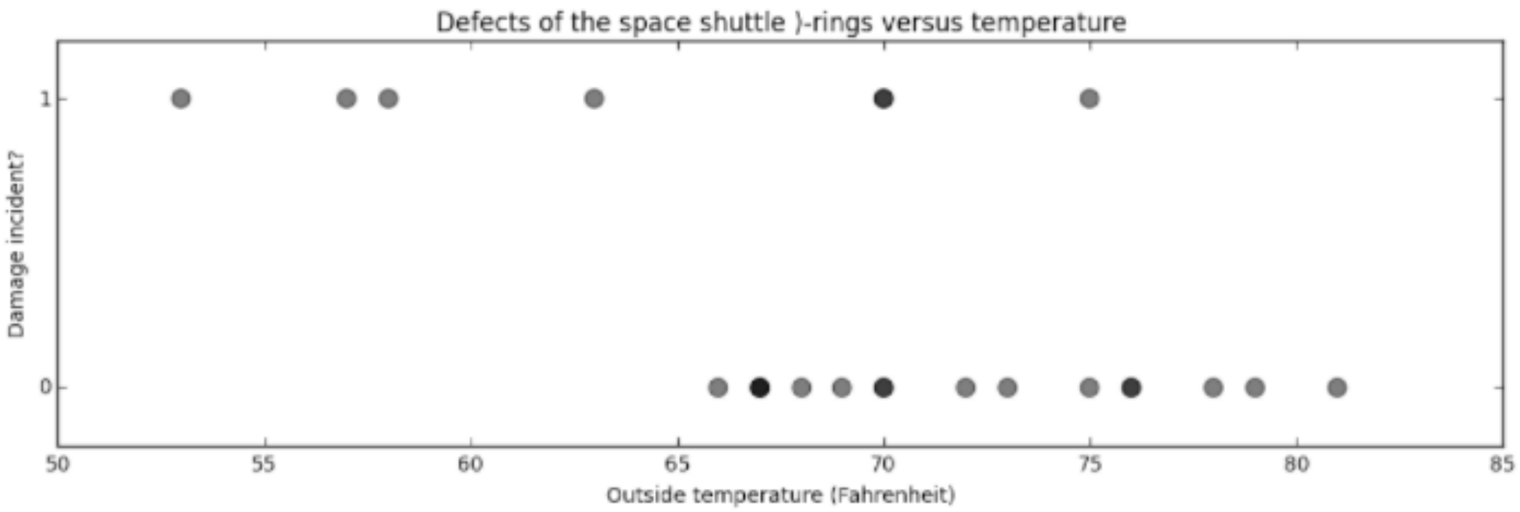
例題：「チャレンジャー号」の悲劇

外気温が下がると、破損発生の確率は明らかに上昇しているが、閾値があるようには見えない

⇛ 問題：「気温tのときの破損発生の確率」を考える

気温が上昇するにつれて、確率が1から0まで変化するはず. . .

⇛ ロジスティック関数をもちいる



ロジスティック関数

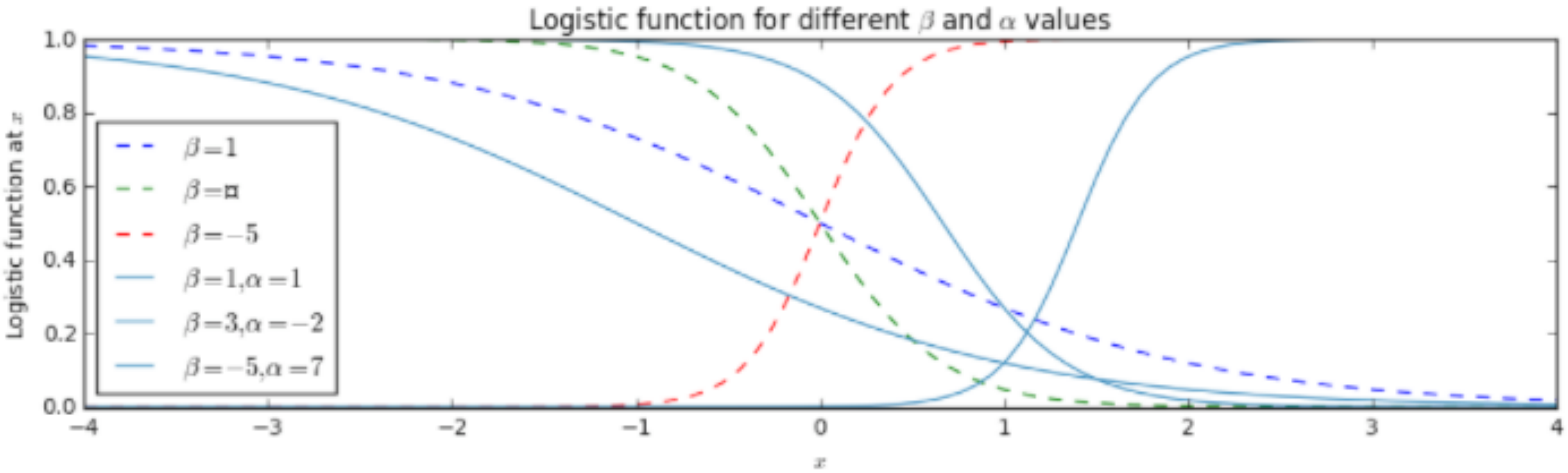
$$p(t) = \frac{1}{1 + e^{\beta t}}$$

```
def logistic(x, beta):
   return 1.0 / (1.0 + np.exp(beta * x))
```

バイアス項を追加

$$p(t) = \frac{1}{1 + e^{\beta t + \alpha}}$$

```
def logistic(x, beta, alpha=0):
   return 1.0/(1.0 + np.exp(np.dot(beta, x) + alpha))
```



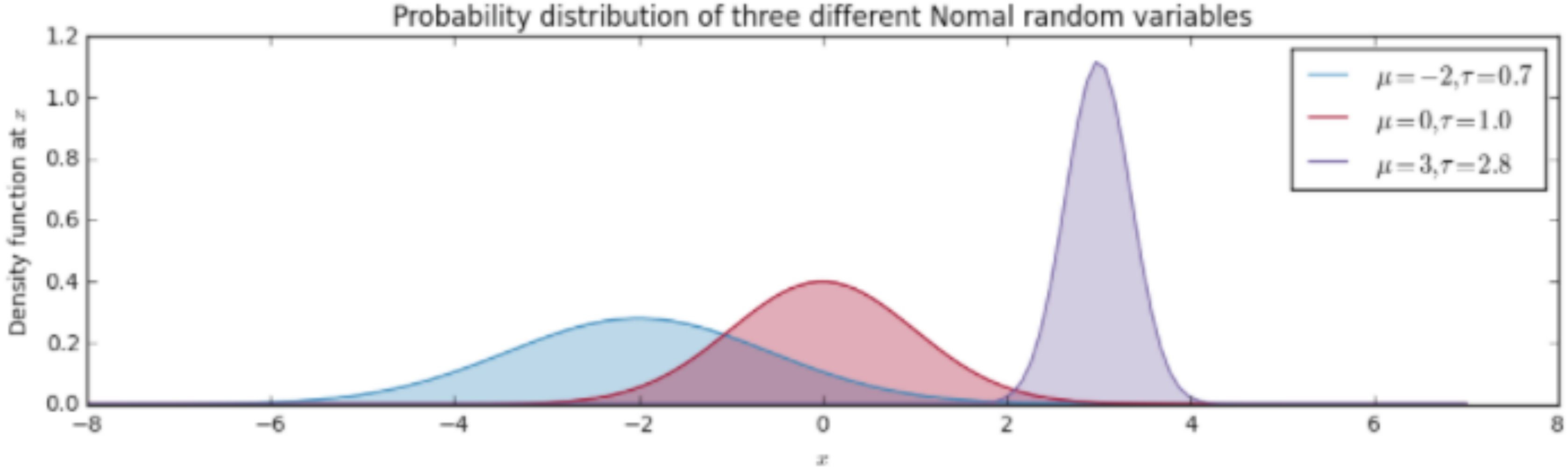
正規分布

平均値の付近に集積するようなデータの**分布**を表した連続的な変数に関する確率**分布**である。

$$f(x|\mu,\tau) = \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau}{2}(x-\mu)^2\right)$$

$$E[X|\mu,\tau] = \mu \qquad \mathrm{Var}(X|\mu,\tau) = \frac{1}{\tau}$$

```
for _mu, _tau, _color in parameters:
    plt.plot(x, nor.pdf(x, _mu, scale=1. / _tau),
             label=r"$\mu = %d, \tau = %.1f$" % (_mu, _tau),
             color=_color)
    plt.fill_between(x, nor.pdf(x, _mu, scale=1. / _tau),
                     color=_color, alpha=.33)
```



例題:「チャレンジャー号」の悲劇 続き

どのように得られた確率と観測データを結びつけるのか？

⇛ ベルヌーイ分布 を用いる

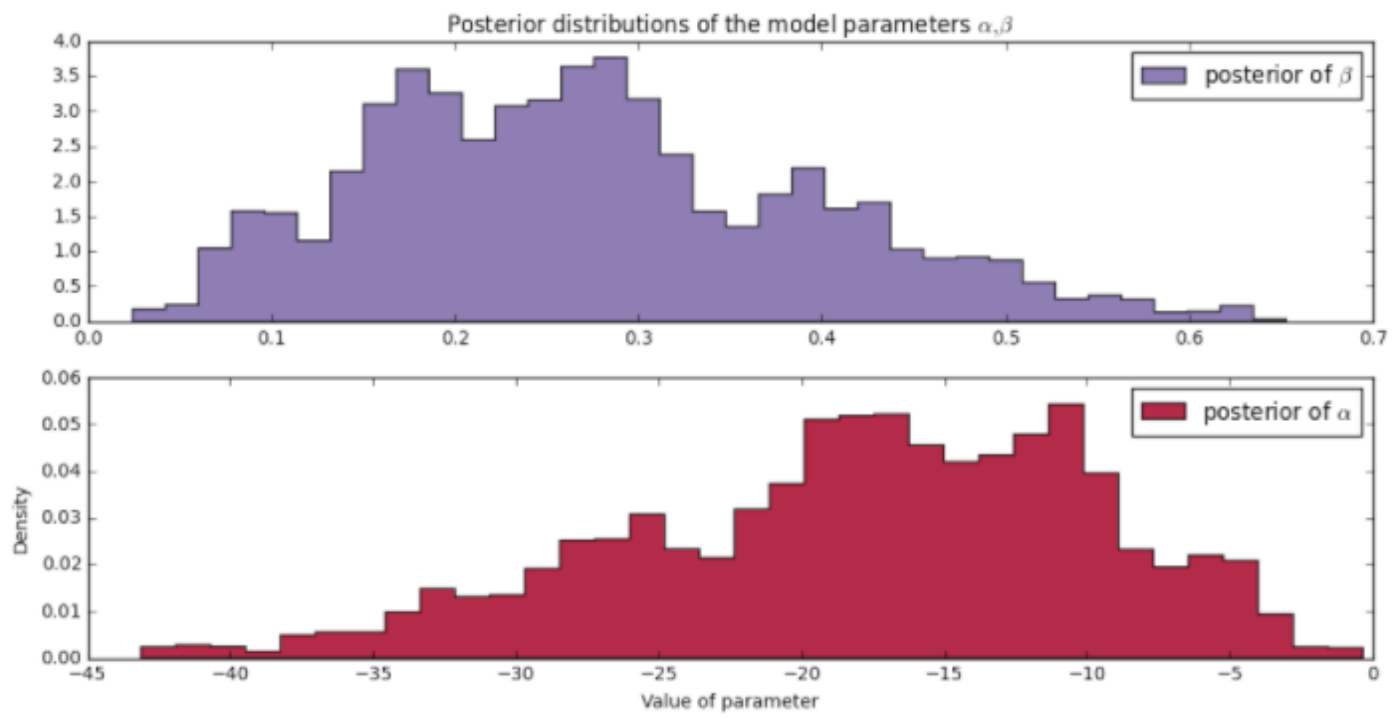
⇛ モデルは $D_i \sim \mathrm{Ber}(p(t_i))$ ($i = 1,\ldots,N$)　(D_i : 破損発生の有無を表す確率変数)

```
# p の確率と観測データをベルヌーイ分布で結びつける
observed = pm.Bernoulli("bernoulli_obs", p, value=D, observed=True)

model = pm.Model([observed, beta, alpha])

# これ以降は3章で
map_ = pm.MAP(model)
map_.fit()
mcmc = pm.MCMC(model)
mcmc.sample(120000, 100000, 2)
```

事後分布からサンプリング

```
%pylab inline
figsize(12.5, 6)

alpha_samples = mcmc.trace('alpha')[:, None] #1次元にする
beta_samples = mcmc.trace('beta')[:, None]

plt.subplot(211)
plt.title("Posterior distributions of "r"the model parameters $\alph
plt.hist(beta_samples, histtype='stepfilled',
      bins=35, alpha=0.85, color="#7A68A6", normed=True,
      label=r"posterior of $\beta$")  # beta の事後分布
plt.legend()

plt.subplot(212)
plt.hist(alpha_samples, histtype='stepfilled',
      bins=35, alpha=0.85, color="#A60628", normed=True,
      label=r"posterior of $\alpha$") # alphaの事後分布
plt.xlabel("Value of parameter") # パラメータ値
plt.ylabel("Density") # 密度
plt.legend()
```

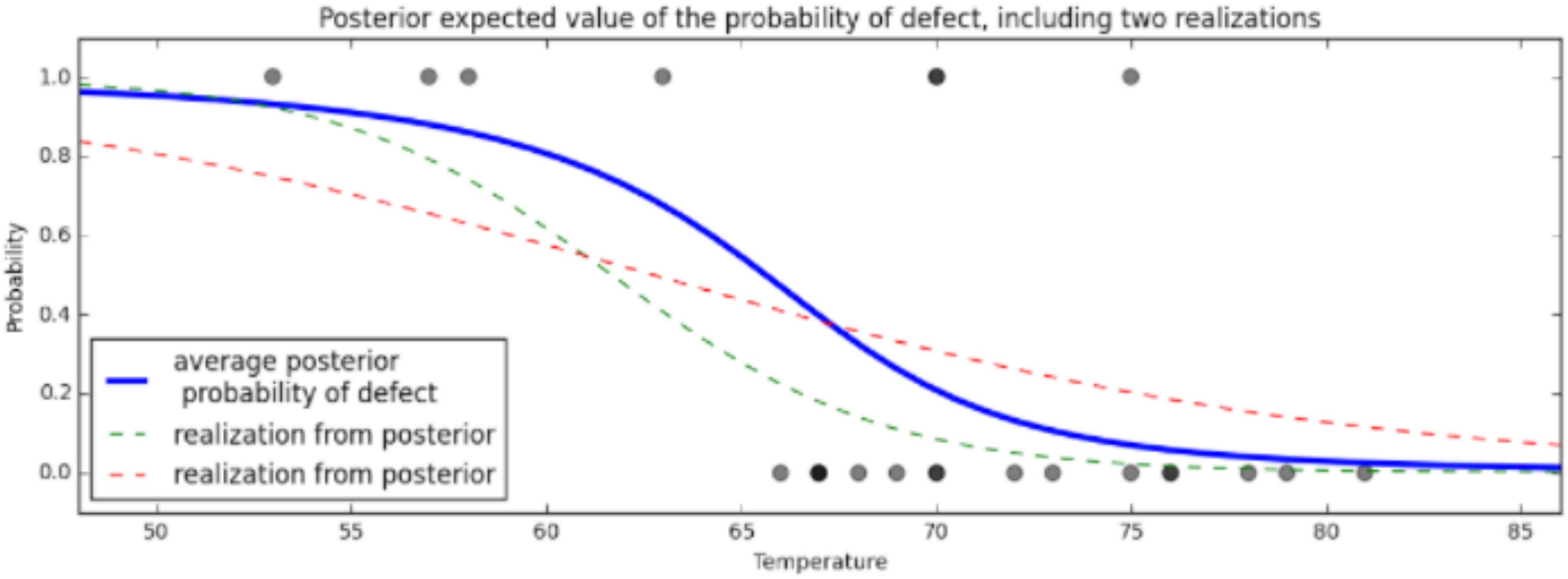
α, βの事後分布をみる



ある外気温についての「期待確率」

p(t_i) がとりそうな値を求める

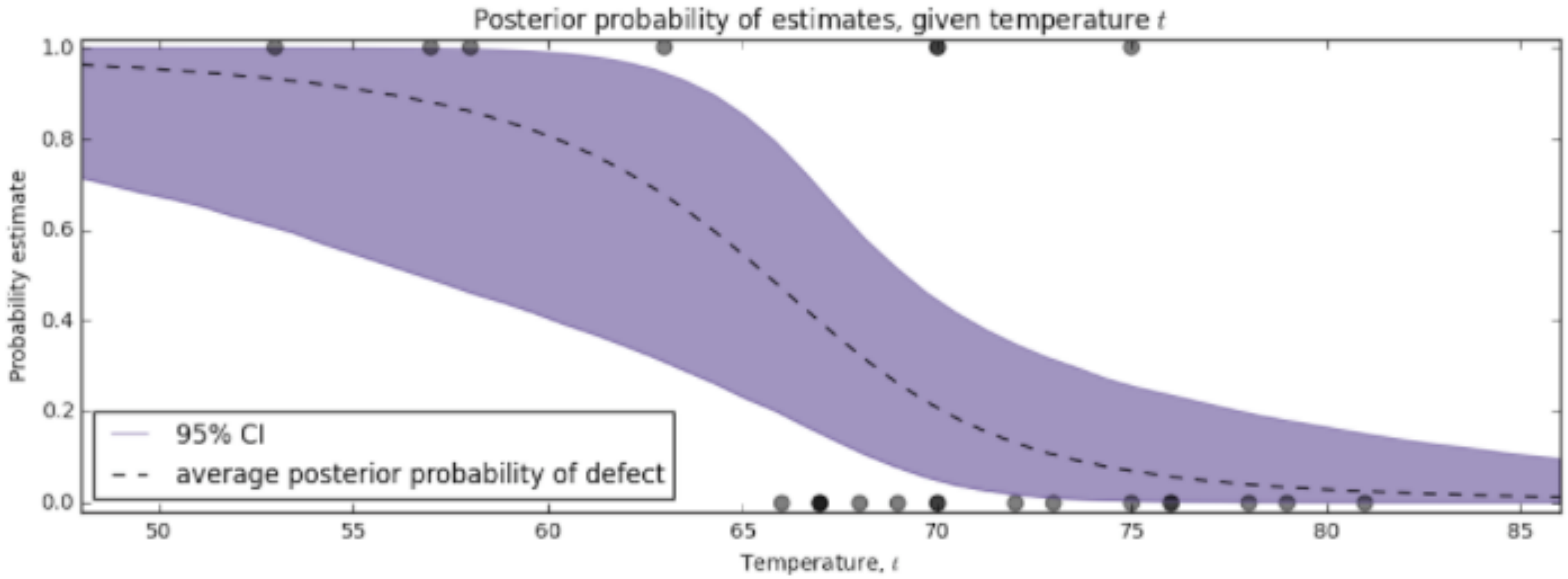
```
p_t = logistic(t.T, beta_samples, alpha_samples)
```



ある外気温についての「期待確率」

95%信頼区間

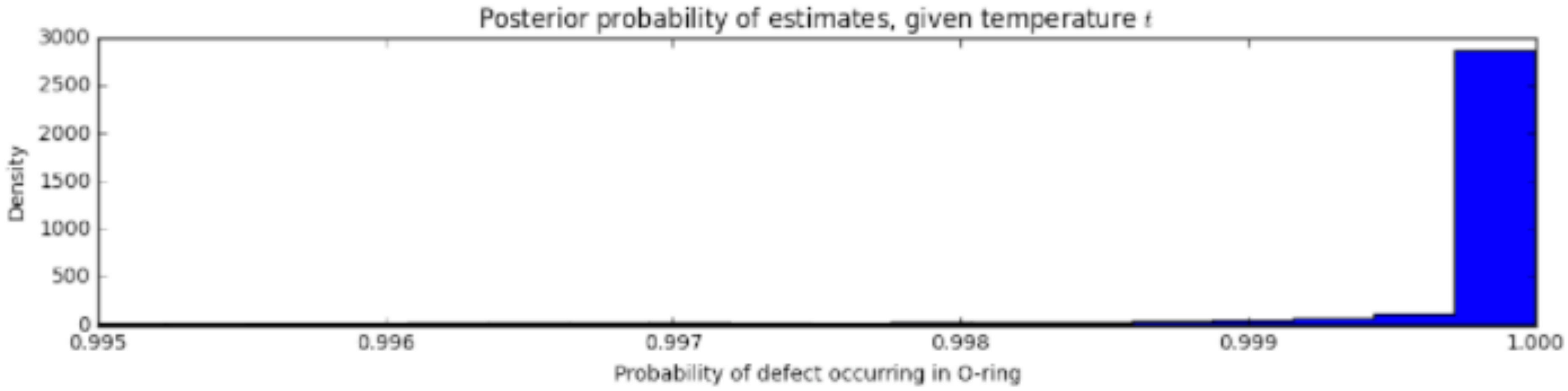
```
# 信頼区間の上下2.5%
qs = mquantiles(p_t, [0.025, 0.975], axis=0)
plt.fill_between(t[:, 0], *qs, alpha=0.7, color="#7A68A6")
```



1986年1月28日 は？？

その日の気温は華氏31度

```
prob_31 = logistic(20, beta_samples, alpha_samples)
```



モデルは適切？

適合度：モデルが適切かどうか測るための指標

モデルがデータと適合しているかどうかをテストする方法
⇛ シミュレーションで人工的にデータを生成して、観測データと比較する

シミュレーションで生成したデータセットが観測データと統計的に似ていなければ、モデルは観測データを正確に表現しているとはいえない

ベイズ的p値 ･･･ モデルの統計的性質を要約した値で、頻度主義のp値に相当

セパレーションプロット

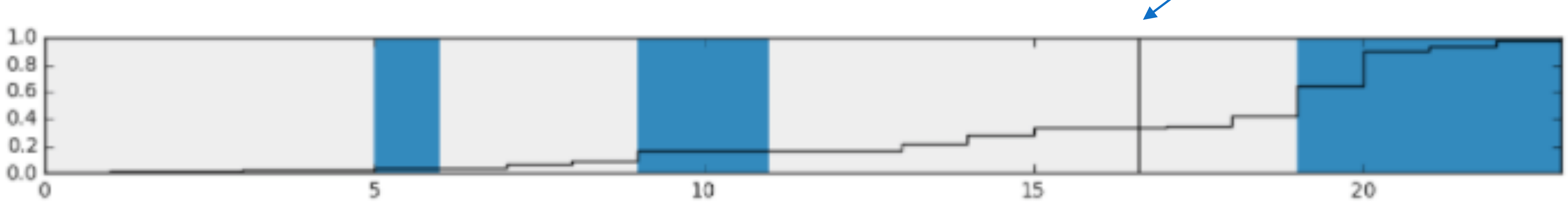
ロジスティック回帰のための新しいデータ可視化手法

Step1 : 各モデルに対し、全てのシミュレーション結果を平均し、ある気温に対して破損と事後分布からサンプルされた回数の割合を計算

Step2 : 事後確率で各列をソート

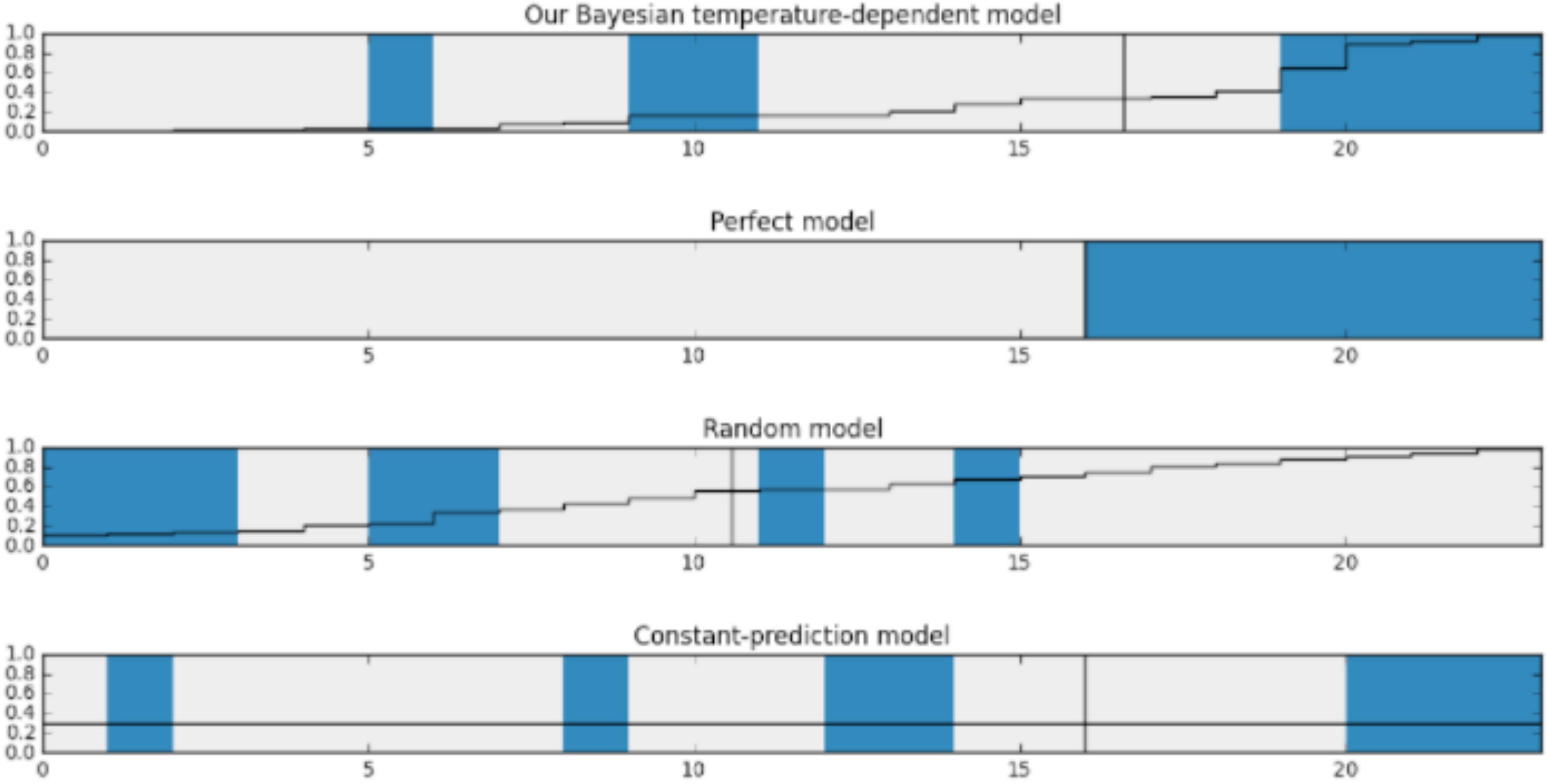
Step3 : プロット

破損発生回数の期待値



青色 : 破損発生
灰色 : 破損なし

セパレーションプロット



演習問題

1. カンニングの例で，観測データ数が極端な場合を考えてみよう．「はい」の回答数が 0，あるいは 100 ならどうなるだろうか？
2. α に対して β をプロットしてみよう．このプロットはどのようになるだろうか？

1はCheating.ipynb
2はChallenger.ipynb
の続きに書いてください！