

医療データ科学実習

Practice of Biomedical Data Science

第7回

# 前回のSlidoの質問に対する回答

**Q1.** 擬似データの作成から棒グラフの作成まで、使っていた変数は「`b_types`」だったのに、プロット作成では「`b_type`」でsが付かなくなるのはなぜでしょうか？sを付けると上手くいかなくなります

**A1.** `b_types`は長さ4のベクトルで、プロット作成の時に指定しているのはデータフレーム`df_normal`の中の列名`b_type`です

**Q2.** 毎回分散と標準偏差を別に指定するのではなく、「今後は `summary` に分散と標準偏差も(別に指示しなくても)表示してね」と指令を出してなおかつ記憶させておくことはできるのでしょうか？

**A2.** これは自作関数を作れば可能ですが、標準的な`summary`と`var`, `sd`関数のみで同様のことをするのは難しという理解です

# 前回のSlidoの質問に対する回答

**Q3.** 連続データの乱数生成のときに、最小値や最大値も指定する事はできるのでしょうか  
(`min`や`max`を試しに入れたらエラーが出ました)

**A3.** 連続データといっても指定する分布はいくつかあります

一様分布でしたら、

```
x <- runif(n = 100, # 100個  
+ min = 10, # 最小値 10  
+ max = 20) # 最大値 20
```

のように`max`と`min`を指定する形で大丈夫です。

正規分布で乱数生成するときは、直接指定することは残念ながらできません。  
例えば...得られた乱数列から端を切り取るか、線形変換して最大値と最小値を調整する形になると思います。(使いたい用途に応じて調整するイメージ)

# 前回のSlidoの質問に対する回答

**Q4.** `Table1`を使用する時に`missing` を記載しない、や`p`値を記載するといった指定をする引数はありますか？また、`missing data`関連で少し思い出したのですが、`if else` を使った時、`na`の値は`else`に入ってしまいますか？`na`のままですか？

**A4.** `p`値の記載は引数 `render.continuous`, `render.categorical` を設定することで可能です（少し複雑ですが）。

`missing`の記載は、`render.missing = NULL`などで対応できます。

`ifelse`関数では`NA`は条件判定の結果が`NA`になるので（`TRUE`, `FALSE`にならない）、`NA`の要素に対する返り値は`NA`になります（結果として変わらない形）。

この要素にアクセスしたい場合は、明示的に`is.na`関数などで`NA`かどうかで条件分岐を先に作る必要があります。

`(x2 <- ifelse(is.na(x1), 欠損値に対する処理, ifelse(欠損のない要素に対する処理))` という形)

# Slidoで質疑応答に参加しよう

医療データ科学実習第2回

医療データ科学実習第2回  
2025/04/15~2025/04/22  
#2974 065

ライブインタラクシ...

Slido を切り替え

ダークモード ☐

Slido について

Slido を無料で試す

Q&A

11 Polls

質問を入力

人気 最近

質問を投稿する

匿名  
6 日前

テスト質問です


Moderator  
6 日前

テスト質問了解です

1 個の質問

0 0


参加  
URL: <https://app.sli.do/event/bFvGHeVHWoLCtNNFL3odmL>



- 匿名で質疑応答に参加できるプラットフォーム
- スマホからでも簡単に利用できます
- 講義後1週間解放しておくので自由に質問やコメントを投稿してください

主催者としてログイン -  
プレゼンテーション モード  
許可可能な使用 - Slido のプライバシー  
Cookie の設定  
© 2012-2025 Slido - 67.29.3

slido



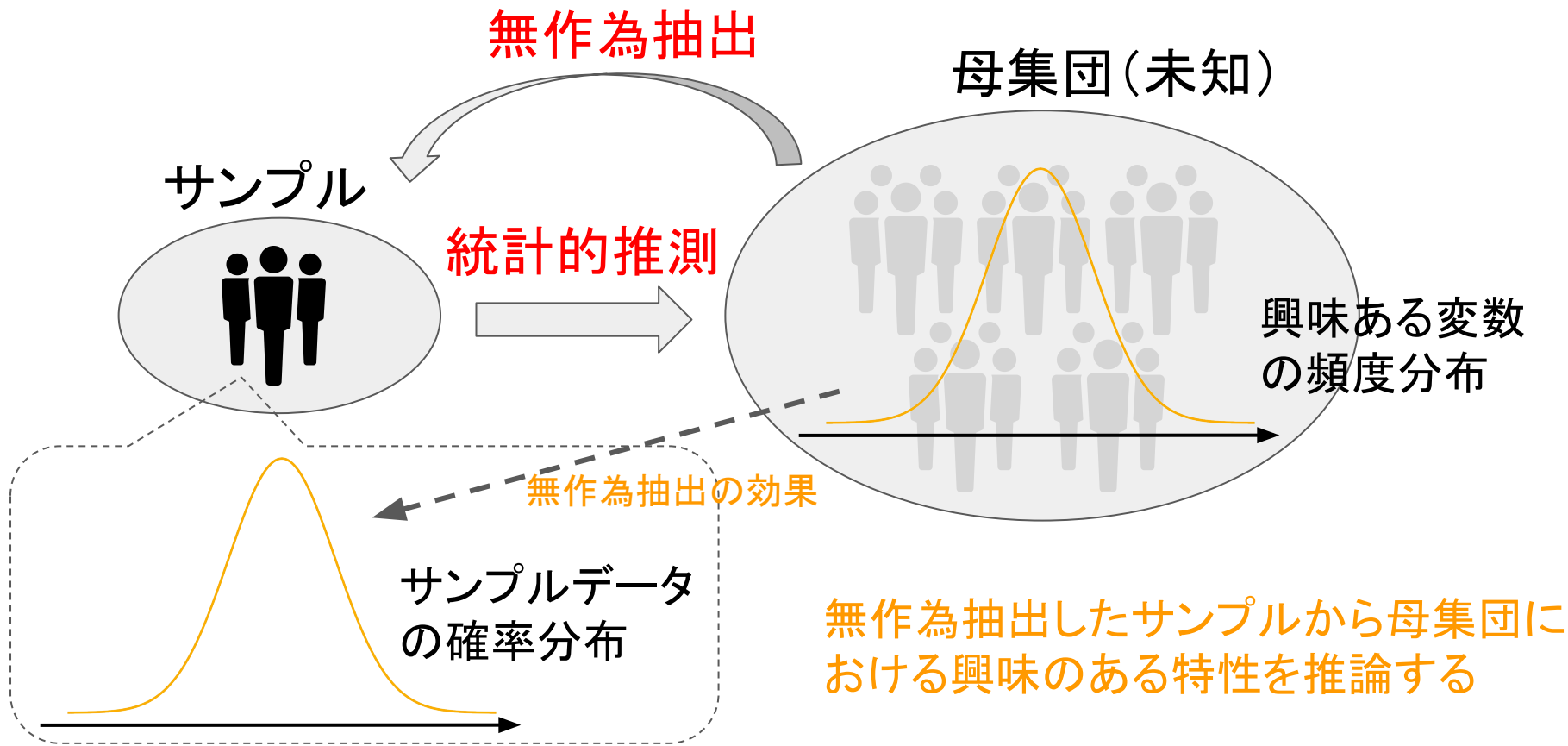
質問投稿部分

人の質問に投票できる  
人気の質問は上位に表示される

# シミュレーションと 統計的推測の基本概念

- 無作為抽出と統計的推測の概要（正規乱数を題材に）
- 反応率の推定と信頼区間，被覆確率
- 反応率の推定における中心極限定理の確認

# 無作為抽出と統計的推測



# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

## サンプルコード

```
# 乱数の個数
n <- 1000

# 標準正規分布から乱数を生成
set.seed(42)
random_numbers <- rnorm(n, mean = 0, sd = 1)

# 確認のため先頭を表示
head(random_numbers)
```



# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

## サンプルコード

```
# 乱数の個数
```

```
n <- 1000
```

生成する擬似乱数の個数（無作為抽出するサンプルの個数）を指定

```
# 標準正規分布から乱数を生成
```

```
set.seed(42)
```

```
random_numbers <- rnorm(n, mean = 0, sd = 1)
```

乱数シードを固定して擬似乱数を1000個生成  
(母集団からサンプルを1000個無作為抽出)

```
# 確認のため先頭を表示
```

```
head(random_numbers)
```

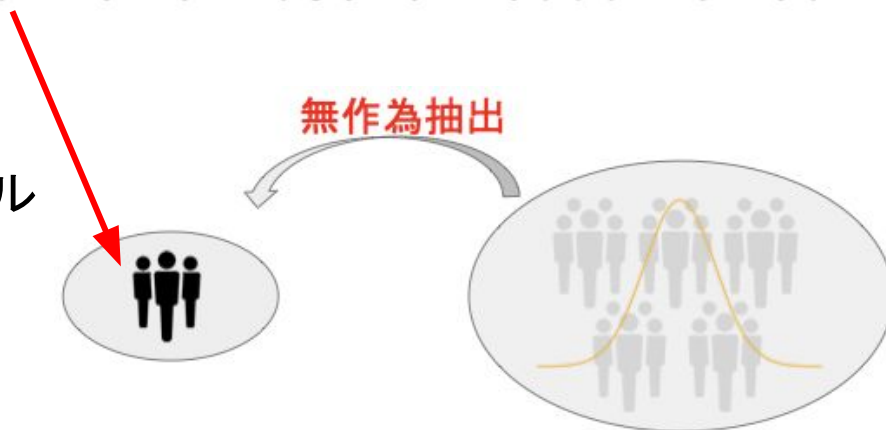
# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

生成されたデータ(先頭の6個):

1.37095844714667 · -0.564698171396089 · 0.363128411337339 ·  
0.63286260496104 · 0.404268323140999 · -0.106124516091484

生成された擬似乱数は  
無作為抽出したサンプル  
データに対応する



この部分のシミュレーション

# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

サンプルコード: サンプルデータの分布を可視化して確認

```
# ヒストグラムでサンプルデータの分布を確認
```

```
par(cex.main = 2.0,  
    cex.lab = 2.0,  
    cex.axis = 2.0)
```

```
hist(random_numbers, breaks = 20, col = "skyblue",  
     main = "標準正規分布から生成した擬似乱数", xlab = "値")
```

# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

サンプルコード: サンプルデータの分布を可視化して確認

# ヒストグラムでサンプルデータの相対頻度を確認

```
par(cex.main = 2.0,  
    cex.lab = 2.0,  
    cex.axis = 2.0)
```

→ プロットの文字サイズを設定する

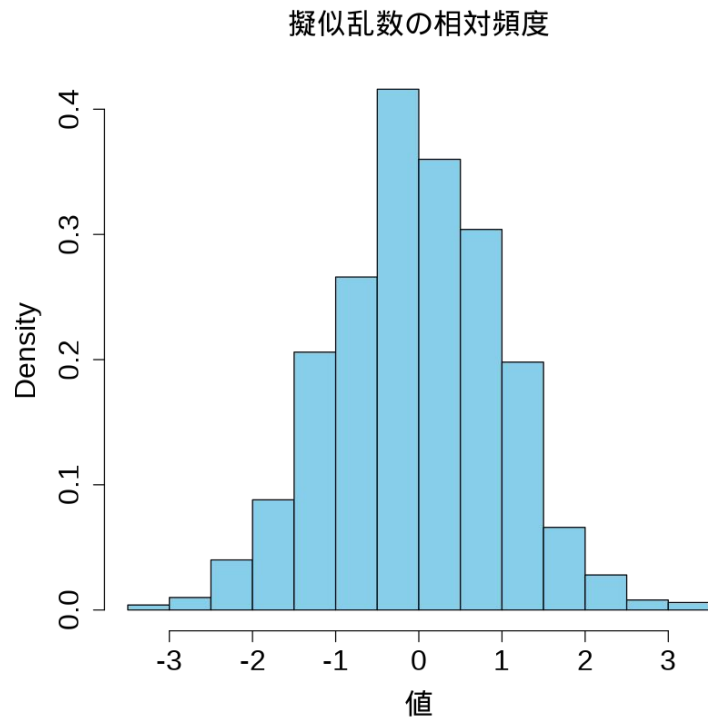
```
hist(random_numbers, breaks = 20, col = "skyblue", prob=TRUE,  
     main = "擬似乱数の相対頻度", xlab = "値")
```

→ 生成した擬似乱数（無作為抽出したサンプルデータ）  
のヒストグラムを描画し、分布の形状を確認

# 無作為抽出のシミュレーション

例題(復習): 標準正規分布からの擬似乱数生成

## 描画されたヒストグラム



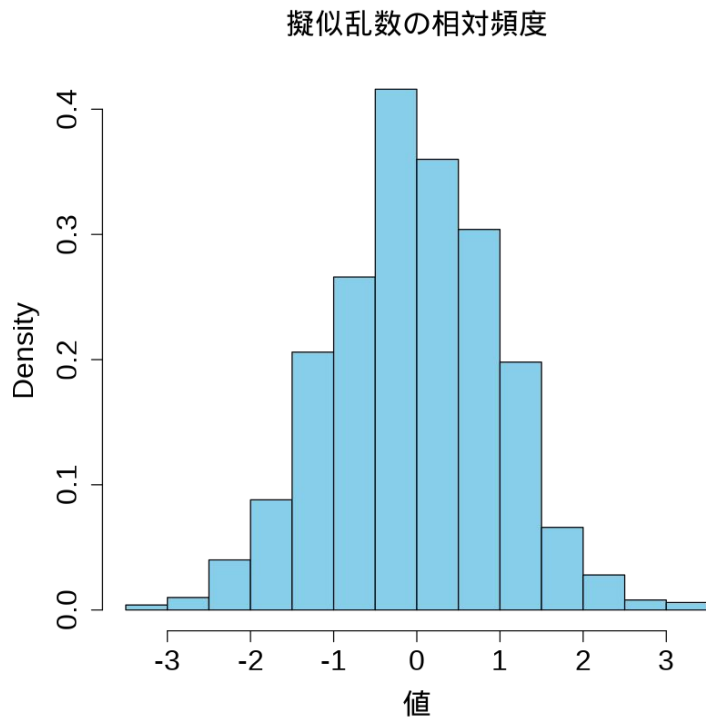
# 無作為抽出のシミュレーション

## 例題(復習): 標準正規分布からの擬似乱数生成

### サンプルコード

ヒストグラムにデータを生成した標準正規分布の密度関数を重ねてみる

```
curve(dnorm(x, mean = 0, sd = 1),  
      col = "red",  
      lwd = 2,  
      add = TRUE)
```



# 無作為抽出のシミュレーション

## 例題(復習): 標準正規分布からの擬似乱数生成

### サンプルコード

ヒストグラムにデータを生成した標準正規分布の密度関数を重ねてみる

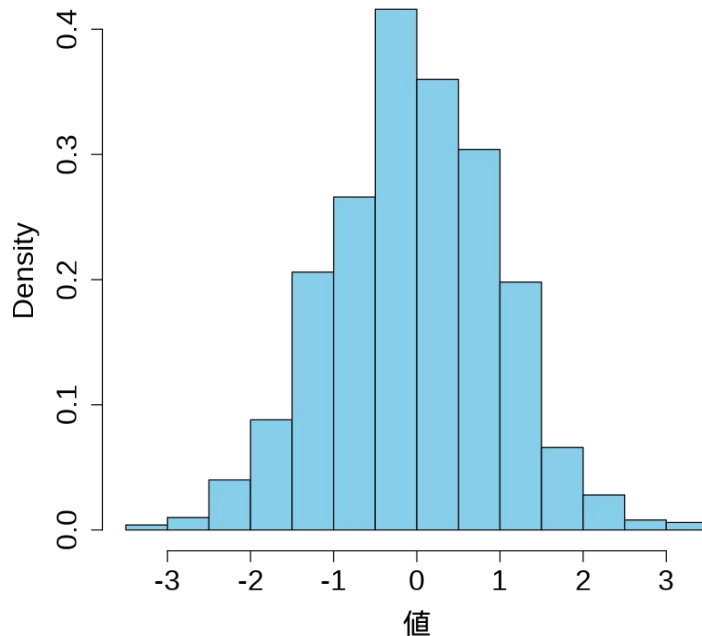
→  $x$ における標準正規分布の密度関数の値を計算する

```
curve(dnorm(x, mean = 0, sd = 1),  
      col = "red",  
      lwd = 2,  
      add = TRUE)
```

→ プロットを重ねる

→ 数式や関数の形を与えてそれをプロット(曲線として描画)する組み込み関数

擬似乱数の相対頻度

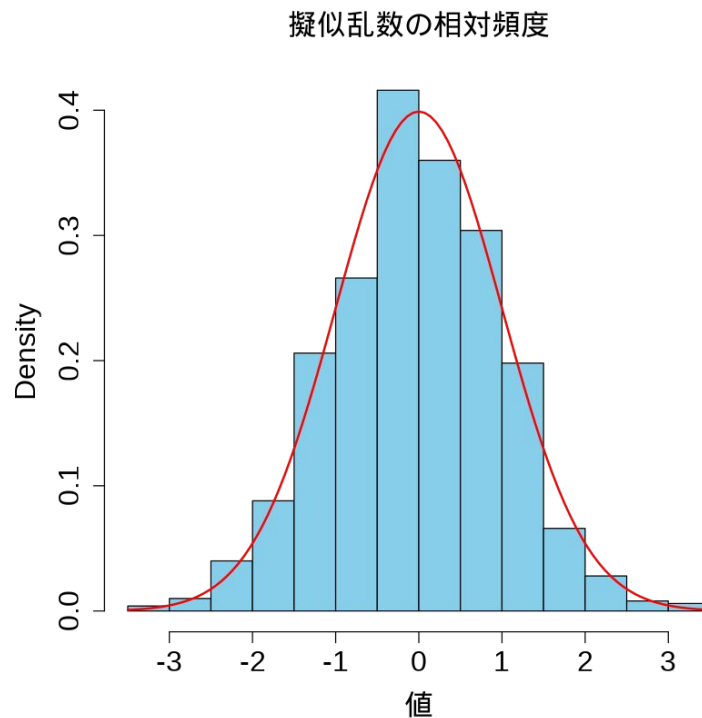


# 無作為抽出のシミュレーション

## 例題(復習): 標準正規分布からの擬似乱数生成

### 描画されたヒストグラム

ヒストグラムにデータを生成した標準正規分布の密度関数を重ねてみる  
→ 母集団の分布(標準正規分布)がサンプルデータの分布として「移植」されていることが確認できる

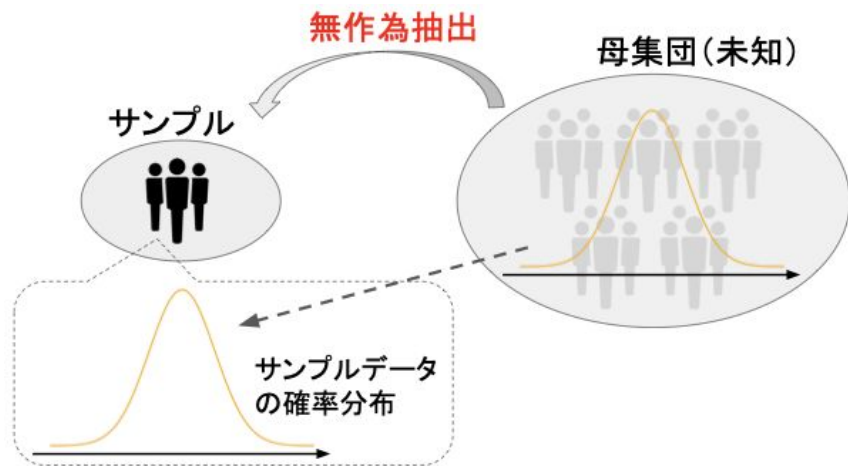




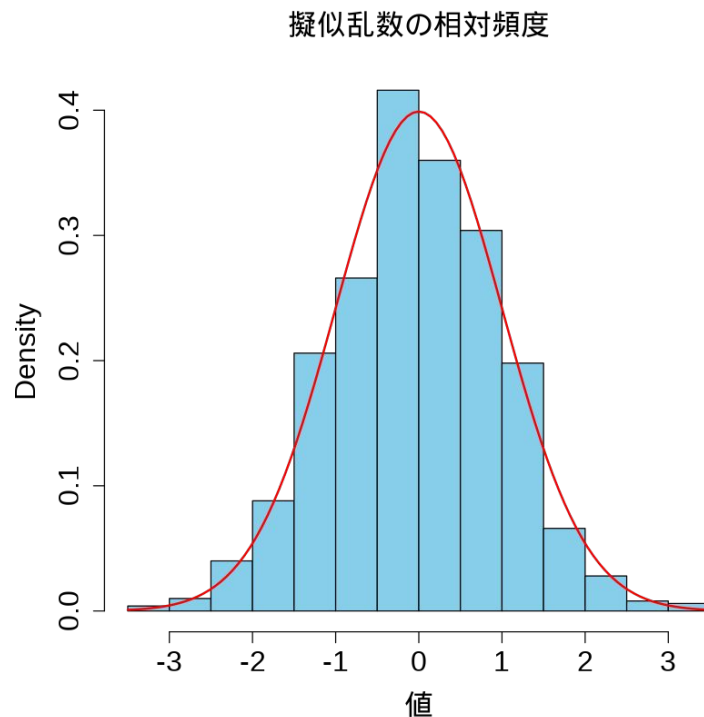
# 無作為抽出のシミュレーション

## 例題(復習): 標準正規分布からの擬似乱数生成

### 描画されたヒストグラム



この部分のシミュレーション



# 統計的推測の基本的な考え方

実際のデータ解析では、

- データ(サンプル, 先の例では生成した乱数)は見える
- 母集団の頻度分布(先の例では標準正規分布)は見えない

## 統計的推測

観測されたデータから母集団の(想定される)分布に関する推論を行う

# 統計的推測のシミュレーション

例題：擬似乱数から生成分布のパラメータを推定する

観測されたデータ(先の擬似乱数生成の結果)：

1.37095844714667 · -0.564698171396089 · 0.363128411337339 ·  
0.63286260496104 · 0.404268323140999 · -0.106124516091484

- このデータの背後にある母集団分布として正規分布を仮定する  
(注：実際に正規乱数なのだが今は見えないフリをする)
- 正規分布は平均と分散という2つのパラメータで特徴付けられる

——→ 平均と分散が決まると対応する正規分布がただ一つに定まる， という意味

# 統計的推測のシミュレーション

例題：擬似乱数から正規分布の平均パラメータを推定する

## 平均の推定に用いる統計量

$$\bar{X} = \frac{X_1 + \cdots + X_n}{n} = \frac{1}{n} \sum_{i=1}^n X_i$$

- 各  $X_i$  は確率変数  $\rightarrow \bar{X}$  も確率変数
- $X_i$  の実現値  $x_i$  (観測データ)を代入することで平均の実現値(推定値)を得る:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

# 統計的推測のシミュレーション

例題: 擬似乱数から正規分布の平均パラメータを推定する

## サンプルコード

```
# 式を直接実装した標本平均の計算
x_mean_1 <- sum(random_numbers)/n

# 組み込み関数による標本平均の計算
x_mean_2 <- mean(random_numbers)

print("式による計算:")
print(x_mean_1)
print("関数による計算:")
print(x_mean_2)
```

# 統計的推測のシミュレーション

例題：擬似乱数から正規分布の平均パラメータを推定する

## サンプルコード

# 式を直接実装した標本平均の計算

```
x_mean_1 <- sum(random_numbers)/n
```

——→ 前ページの推定値の式を直接実装して計算

# 組み込み関数による標本平均の計算

```
x_mean_2 <- mean(random_numbers)
```

——→ 標本平均を計算するRの組み込み関数

```
print("式による計算：")
```

```
print(x_mean_1)
```

```
print("関数による計算：")
```

```
print(x_mean_2)
```

——→ print文で計算結果  
を表示

# 統計的推測のシミュレーション

例題：擬似乱数から正規分布の平均パラメータを推定する

計算結果

[1] "式による計算："

[1] -0.02582443

[1] "関数による計算："

[1] -0.02582443

- どちらの方法で計算しても同じ結果となった(計算が正しく行われていることの確認方法の一つ)
- 得られた推定値は真の平均(0)に近い → 良さそう  
Q. この推定は本当に信用できるものなのか？

# 大数の法則と中心極限定理

## 大数の法則

独立に同一の分布に従うデータの標本平均は、真の平均に収束する

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \rightarrow \mu \quad (n \rightarrow \infty)$$



# 大数の法則と中心極限定理

## 大数の法則

独立に同一の分布に従うデータの標本平均は、真の平均に収束する

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

標本平均

矢印の左側が右側に近づいていくことを表す



$\mu$

真の平均

$$(n \rightarrow \infty)$$

サンプルの数を増やすほど標本平均と真の平均が近づくことを表す

→ データ数が増えるにつれて標本平均による推定の精度が上がっていくイメージ

# 大数の法則と中心極限定理

## 大数の法則

データ数を 1000, 10000, 100000, 1000000 と増やしてみよう

n=1000

[1] "式による計算 : "

[1] -0.02582443

[1] "関数による計算 : "

[1] -0.02582443

n=10000

[1] "式による計算 : "

[1] -0.01130945

[1] "関数による計算 : "

[1] -0.01130945

n=100000

[1] "式による計算 : "

[1] -0.004126182

[1] "関数による計算 : "

[1] -0.004126182

n=1000000

[1] "式による計算 : "

[1] 0.0005737398

[1] "関数による計算 : "

[1] 0.0005737398

\*パラメータ  
の真値 = 0

# 大数の法則と中心極限定理

## 中心極限定理

独立に同一の分布(平均  $\mu$  分散  $\sigma^2$ とする)に従うデータの標本平均は, データ数  $n$  が十分大きいとき近似的に, 平均が  $\mu$ , 分散が  $\sigma^2/n$  の正規分布に従う

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \sim \mathcal{N}(\mu, \sigma^2/n) \quad (n \rightarrow \infty)$$

# 大数の法則と中心極限定理

## 中心極限定理

独立に同一の分布(平均  $\mu$  分散  $\sigma^2$ とする)に従うデータの標本平均は、データ数  $n$  が十分大きいとき近似的に、平均が  $\mu$ 、分散が  $\sigma^2/n$  の正規分布に従う

左側の確率変数が右側の確率分布に従うことを表す

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \sim \mathcal{N}(\mu, \sigma^2/n)$$

正規分布を表す記号

$$(n \rightarrow \infty)$$

データ数が多いほど左の近似精度が上がっていくことを表す

# 大数の法則と中心極限定理

## 中心極限定理

乱数を繰り返し生成する試行を2000回行い、標本平均のヒストグラムを描画してみよう

サンプルコード ステップ1: 標本サイズと試行回数を指定

```
# パラメータ設定
```

```
n <- 1000      # 1回あたりの標本サイズ
```

```
trials <- 2000 # 試行回数
```

```
# 標本平均を保存するベクトルを用意
```

```
sample_means <- numeric(trials)
```

# 大数の法則と中心極限定理

## 中心極限定理

乱数を繰り返し生成する試行を2000回行い、標本平均のヒストグラムを描画してみよう

サンプルコード ステップ1: 標本サイズと試行回数を指定

```
# パラメータ設定
```

```
n <- 1000      # 1回あたりの標本サイズ
```

```
trials <- 2000  # 試行回数
```

```
# 標本平均を保存するベクトルを用意
```

```
sample_means <- numeric(trials)
```

→ 試行回数次元の空のベクトルを作成する

# 大数の法則と中心極限定理

## 中心極限定理

乱数を繰り返し生成する試行を2000回行い、標本平均のヒストグラムを描画してみよう

サンプルコード ステップ2: 2000回の繰り返し試行

```
# 試行を繰り返し実行
```

```
for (i in 1:trials) {
```

```
  random_sample <- rnorm(n, mean = 0, sd = 1)
```

```
  sample_means[i] <- mean(random_sample)
```

```
}
```

# 大数の法則と中心極限定理

## 中心極限定理

乱数を繰り返し生成する試行を2000回行い、標本平均のヒストグラムを描画してみよう

サンプルコード ステップ2: 2000回の繰り返し試行

```
# 試行を繰り返し実行
```

```
for (i in 1:trials) {
```

```
  random_sample <- rnorm(n, mean = 0, sd = 1)
```

```
  sample_means[i] <- mean(random_sample)
```

```
} 標準正規乱数を生成し、標本平均を計算して i 番目の位置に格納する
```

{ } 内の処理を trials 回繰り返すループ

i は処理の実行回数のカウンタで、i=trials になったらループを抜ける



# 大数の法則と中心極限定理

## 中心極限定理

乱数を繰り返し生成する試行を2000回行い、標本平均のヒストグラムを描画してみよう

サンプルコード ステップ3: 標本平均のヒストグラムを描画

```
# 標本平均のヒストグラムを描画
```

```
par(cex.main = 2.0,  
    cex.lab = 2.0,  
    cex.axis = 2.0)
```

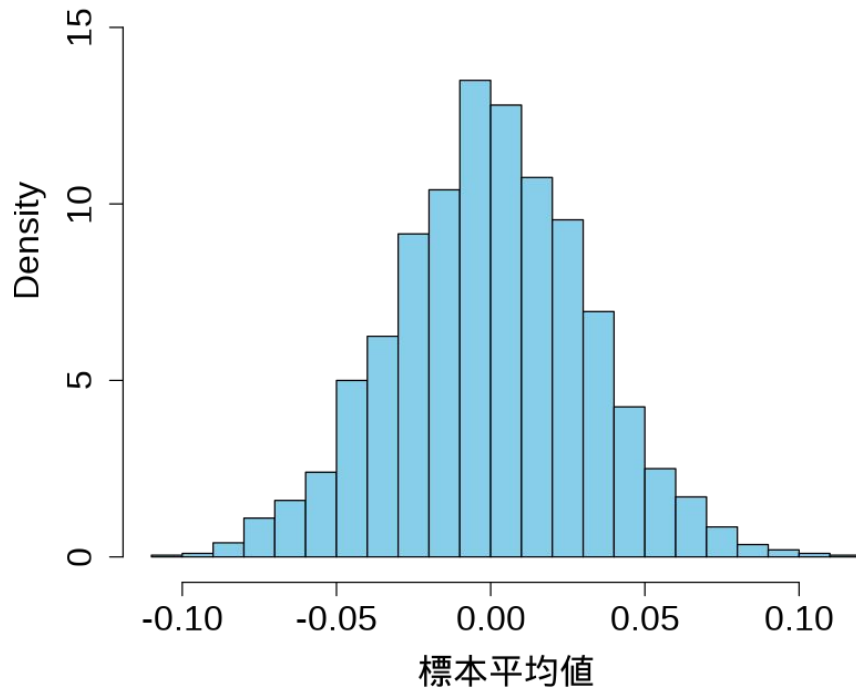
```
hist(sample_means, breaks = 30, col = "skyblue", prob=TRUE,  
     main = "2000回の試行で得られた標本平均の相対頻度",  
     xlab = "標本平均値")
```

# 大数の法則と中心極限定理

## 中心極限定理

### 描画されたヒストグラム

2000回の試行で得られた標本平均の相対頻度



# 大数の法則と中心極限定理

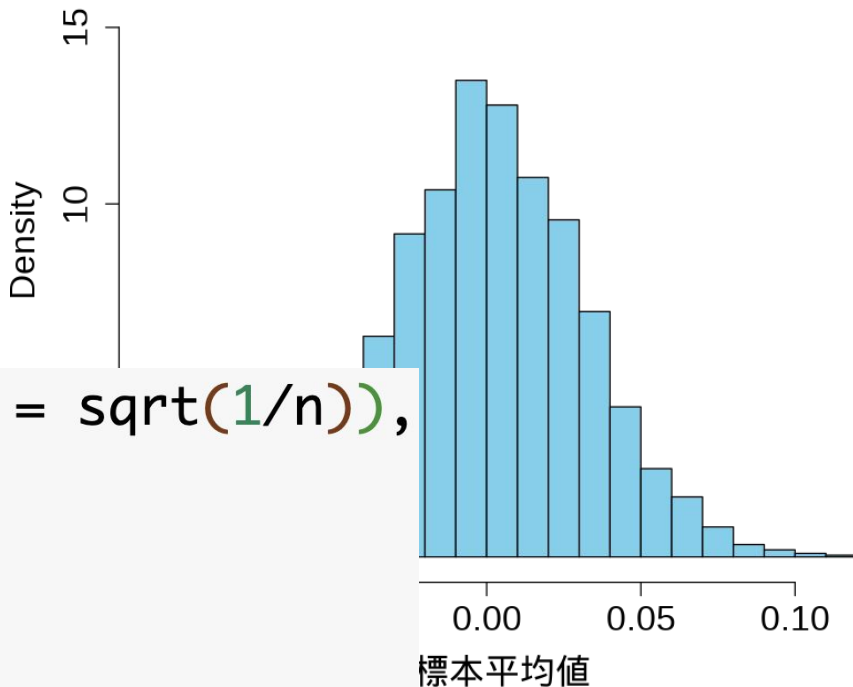
## 中心極限定理

2000回の試行で得られた標本平均の相対頻度

### 描画されたヒストグラム

平均が0, 分散が $1/n$ の正規分布の密度関数を重ねてプロットしてみる

```
curve(dnorm(x, mean = 0, sd = sqrt(1/n)),  
      col = "red",  
      lwd = 2,  
      add = TRUE)
```



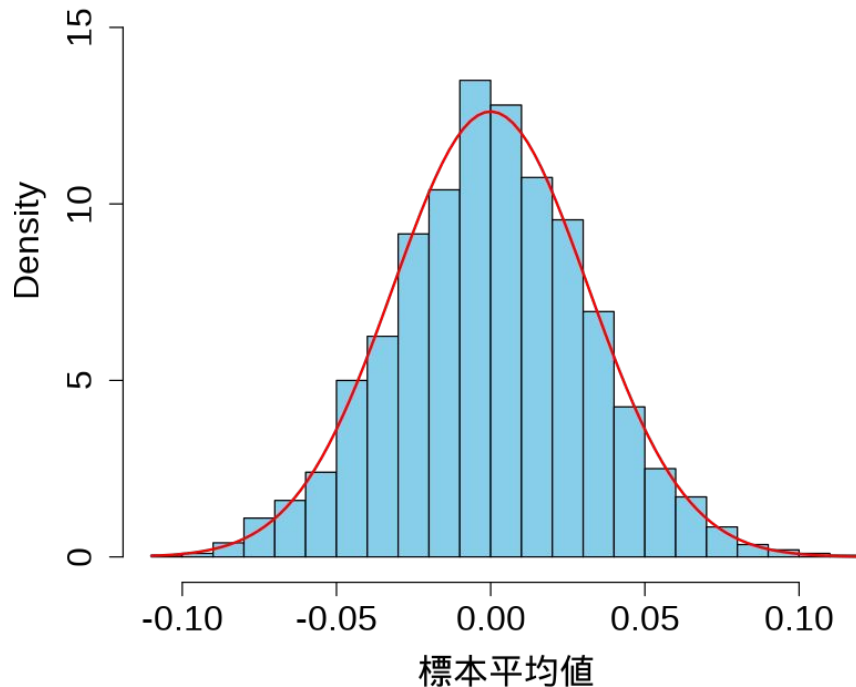
# 大数の法則と中心極限定理

## 中心極限定理

### 描画されたヒストグラム

中心極限定理の主張と良く  
整合するようなプロットが得  
られた

2000回の試行で得られた標本平均の相対頻度



# 反応率の予測問題

例題:

- がん第2相単群臨床試験(新規治療の有効性を評価)
- 主要評価項目は腫瘍の縮小反応の有無

設定: データ  $X$  は被験者が縮小反応を示したかどうか(2値)

- 被験者数: 20名
- 縮小反応を示した人数: 11名
- 薬剤の真の反応率(実際には未知): 0.4

# 反応率の予測問題

設定: データ  $X$  は被験者が縮小反応を示したかどうか(2値)

- 被験者数: 20名
- 縮小反応を示した人数: 11名
- 薬剤の真の反応率(実際には未知): 0.4

**演習**: RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算を試みよう

# 反応率の予測問題

設定: データ  $X$  は被験者が縮小反応を示したかどうか (2値)

- 被験者数: 20名
- 縮小反応を示した人数: 11名
- 薬剤の真の反応率 (実際には未知):  $p = 0.4$

## 演習のヒント (松井先生の医療データ科学のスライド)

- 反応率は **反応した人数 / 全被験者数** で推定できる →  
データの標本平均を計算するのと同じ
- 2値の事象 (反応の有無) では分散は  $p(1 - p)$  →  
標本平均の分散は  $p(1 - p) / n$  で計算できる
- 標本平均の標準誤差は分散の平方根で計算できる

# 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

サンプルコード ステップ1: 実験データの用意

```
# データ数
n <- 20
# 観測データをシミュレート (1が11個、0が9個)
vec <- c(rep(1, 11), rep(0, 9))
x <- sample(vec)
print(x)
```



# 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

サンプルコード ステップ1: 実験データの用意

```
# データ数
```

```
n <- 20
```

```
# 観測データをシミュレート (1が11個、0が9個)
```

```
vec <- c(rep(1, 11), rep(0, 9))
```

1が11個, 0が9個順番に  
並んだベクトルを作成

```
x <- sample(vec)
```

ベクトルの要素をランダム  
に並べ替え

```
print(x)
```

## 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

作成した20人分の反応の有無データ

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

# 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

## サンプルコード ステップ2: 統計量の計算

```
# 反応率, 分散, 標準誤差の計算
```

```
p_est <- sum(x) / n
```

```
p_var_est <- p_est * (1 - p_est) / n
```

```
p_se_est <- sqrt(p_var_est)
```

```
print(paste("反応率の推定値 :", p_est))
```

```
print(paste("分散の推定値 :", p_var_est))
```

```
print(paste("標準誤差の推定値 :", p_se_est))
```

# 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

## サンプルコード ステップ2: 統計量の計算

# 反応率, 分散, 標準誤差の計算

```
p_est <- sum(x) / n  
p_var_est <- p_est * (1 - p_est) / n  
p_se_est <- sqrt(p_var_est)
```

数式に基づいてそれぞれの統計量を計算

```
print(paste("反応率の推定値 :", p_est))  
print(paste("分散の推定値 :", p_var_est))  
print(paste("標準誤差の推定値 :", p_se_est))
```

計算結果を表示

## 反応率の予測問題

**演習** : RStudioで上の情報から反応率, 分散, 標準誤差の推定値の計算をしてみよう

### 計算結果

[1] "反応率の推定値 : 0.55"

[1] "分散の推定値 : 0.012375"

[1] "標準誤差の推定値 : 0.111242977306435"

# 反応率の推定値のばらつきの評価

**演習** : RStudioで

- 被験者数は20名
- 真の反応率は0.4

の設定でデータ生成, 反応率の計算を2000回繰り返すシミュレーションを行い, 反応率の推定値のヒストグラムを描いてみよう

# 反応率の推定値のばらつきの評価

## サンプルコード ステップ1: 2000回のシミュレーション

```
# 2000回のシミュレーション
set.seed(42) # 乱数シードを固定
trials <- 2000 # シミュレーションの回数
n <- 20 # データ数
p <- 0.4 # 真の反応率
p_est_vec <- numeric(trials)
for(i in 1:trials){
  observation <- rbinom(n, size = 1, prob = p)
  p_est_vec[i] <- sum(observation) / n
}
print(p_est_vec)
```

# 反応率の推定値のばらつきの評価

## サンプルコード ステップ1: 2000回のシミュレーション

```
# 2000回のシミュレーション
```

```
set.seed(42) # 乱数シードを固定
```

```
trials <- 2000 # シミュレーションの回数
```

```
n <- 20 # データ数
```

```
p <- 0.4 # 真の反応率
```

問題設定のパラメータを  
定義

```
p_est_vec <- numeric(trials)
```

反応率の推定結果を格納する空ベクトル

```
for(i in 1:trials){
```

```
  observation <- rbinom(n, size = 1, prob = p)
```

→ 20人分の反応の有  
無データを1回作成

```
  p_est_vec[i] <- sum(observation) / n
```

```
}
```

```
print(p_est_vec)
```

データを生成し、反応率を計算して保存する繰り返し処理



# 反応率の推定値のばらつきの評価

## 反応率のシミュレーション結果

[1]	0.55	0.55	0.45	0.35	0.40	0.50	0.60	0.45	0.30	0.45	0.30	0.25	0.15	0.35
[15]	0.25	0.35	0.35	0.35	0.45	0.35	0.25	0.20	0.50	0.35	0.30	0.20	0.45	0.45
[29]	0.35	0.45	0.25	0.30	0.35	0.30	0.55	0.50	0.45	0.35	0.50	0.35	0.35	0.25
[43]	0.35	0.40	0.55	0.30	0.50	0.40	0.30	0.30	0.30	0.30	0.45	0.35	0.40	0.60
[57]	0.35	0.35	0.30	0.35	0.45	0.45	0.60	0.45	0.35	0.40	0.50	0.30	0.25	0.15
[71]	0.25	0.50	0.50	0.35	0.35	0.35	0.50	0.45	0.40	0.15	0.60	0.40	0.30	0.40
[85]	0.35	0.45	0.40	0.15	0.35	0.35	0.30	0.50	0.65	0.50	0.35	0.60	0.50	0.45
[99]	0.50	0.50	0.60	0.20	0.55	0.35	0.30	0.60	0.30	0.50	0.30	0.50	0.30	0.50
[113]	0.30	0.45	0.45	0.65	0.45	0.60	0.10	0.75	0.55	0.50	0.40	0.35	0.45	0.40

各要素が独立な試行における反応率の推定値であるような  
2000次元のベクトルができる

# 反応率の推定値のばらつきの評価

## サンプルコード ステップ2: 反応率のヒストグラム描画

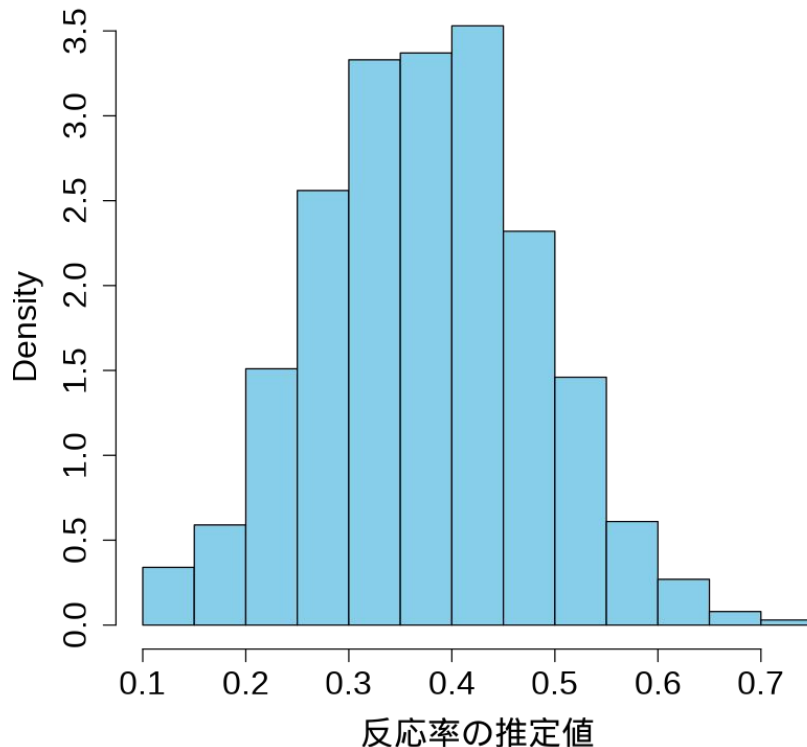
```
hist(p_est_vec, breaks = 10, col = "skyblue", prob=TRUE,  
     main = "2000回の試行で得られた反応率の相対頻度",  
     xlab = "反応率の推定値")
```

# 反応率の推定値のばらつきの評価

## 描画されたヒストグラム

真の反応率  $p = 0.4$  の  
まわりに推定値がばらつ  
くような結果が得られた

2000回の試行で得られた反応率の相対頻度



## 反応率の推定値のばらつきの評価

**演習**: いまのシミュレーションで、反応率の経験分散と理論分散が整合しているかを確認しよう

### サンプルコード

得られた2000個の反応率の分散を計算して理論値と比較

```
print(p*(1-p)/n)
print(var(p_est_vec)*(trials-1)/trials)
```

## 反応率の推定値のばらつきの評価

**演習**: いまのシミュレーションで、反応率の経験分散と理論分散が整合しているかを確認しよう

### サンプルコード

得られた2000個の反応率の分散を計算して理論値と比較

分散の理論値を計算

```
print(p*(1-p)/n)
```

```
print(var(p_est_vec)*(trials-1)/trials)
```

シミュレーション結果から経験分散を計算

- `var()` 関数は不偏分散を計算する (n-1で割るやつ)
- 標本分散を計算するために、`var`の結果に  $n-1/n$  を掛けている

## 反応率の推定値のばらつきの評価

**演習** : いまのシミュレーションで、反応率の経験分散と理論分散が整合しているかを確認しよう

### 分散の計算結果

[1] "分散の理論値 : 0.012"

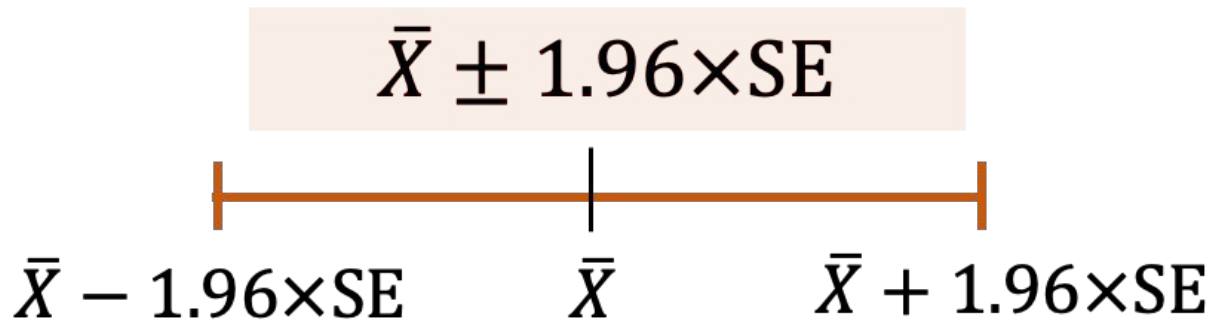
[1] "シミュレーション結果から計算した経験分散 0.011571219375"

おおむね理論値に近い分散が得られている

## 反応率の推定値の信頼区間

**演習** : いまのシミュレーションで, 反応率の推定値に対する  
(正規近似のもとでの) 95% 信頼区間を計算しよう

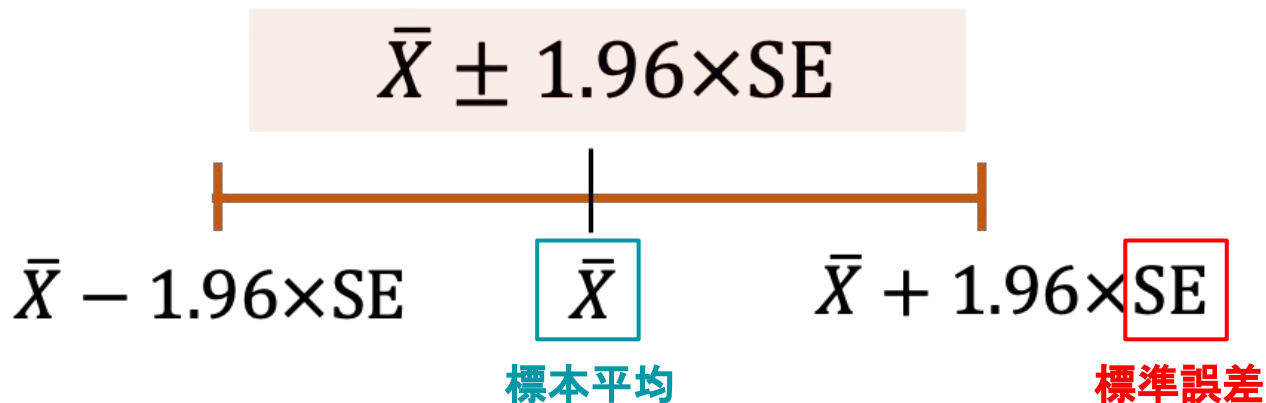
(復習) 統計量の95%信頼区間 (cf: 松井先生のスライド)



## 反応率の推定値の信頼区間

**演習** : いまのシミュレーションで, 反応率の推定値に対する  
(正規近似のもとでの) 95% 信頼区間を計算しよう

(復習) 統計量の95%信頼区間 (cf: 松井先生のスライド)





# 反応率の推定値の信頼区間

**演習** : いまのシミュレーションで, 反応率の推定値に対する  
(正規近似のもとでの) 95% 信頼区間を計算しよう

## サンプルコード (1回の試行)

```
n <- 20 # データ数
p <- 0.4 # 真の反応率
set.seed(42)
observation <- rbinom(n, size = 1, prob = p)
p_est <- sum(observation) / n # 反応率の推定値
p_est_se <- sqrt(p_est*(1 - p_est)/n) # 標準誤差
print(paste("反応率の推定値 :", p_est))
print(paste("反応率の95%信頼区間の下限 :", p_est - 1.96*p_est_se))
print(paste("反応率の95%信頼区間の上限 :", p_est + 1.96*p_est_se))
```

# 反応率の推定値の信頼区間

**演習**:いまのシミュレーションで, 反応率の推定値に対する  
(正規近似のもとでの)95%信頼区間を計算しよう

## サンプルコード(1回の試行)

```
n <- 20 # データ数  
p <- 0.4 # 真の反応率
```

```
set.seed(42)
```

```
observation <- rbinom(n, size = 1, prob = p)
```

乱数シードを固定して反応の  
有無データを20人分生成

```
p_est <- sum(observation) / n # 反応率の推定値
```

```
p_est_se <- sqrt(p_est*(1 - p_est)/n) # 標準誤差
```

```
print(paste("反応率の推定値:", p_est))
```

```
print(paste("反応率の95%信頼区間の下限:", p_est - 1.96*p_est_se))
```

```
print(paste("反応率の95%信頼区間の上限:", p_est + 1.96*p_est_se))
```

## 反応率の推定値の信頼区間

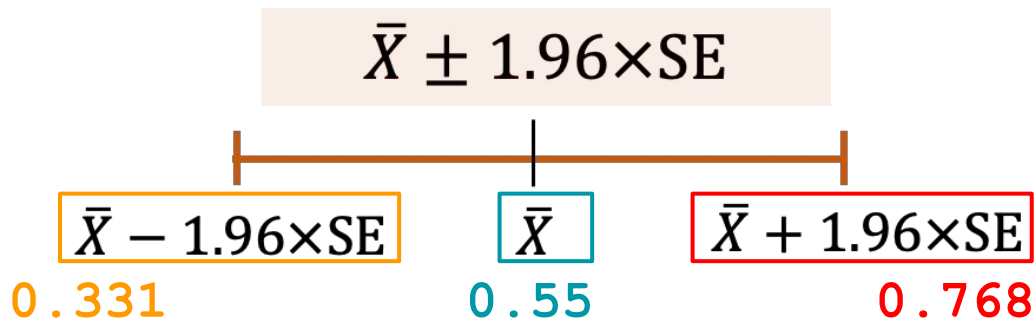
**演習** : いまのシミュレーションで、反応率の推定値に対する  
(正規近似のもとでの)95%信頼区間を計算しよう

### 実行結果

[1] "反応率の推定値 : 0.55"

[1] "反応率の95%信頼区間の下限 : 0.331963764479388"

[1] "反応率の95%信頼区間の上限 : 0.768036235520613"



## 反応率の信頼区間の被覆確率

**演習** : 反応率の推定値に対する95%信頼区間を計算するシミュレーションを1000回行い, **被覆確率**を求めよう

**信頼区間がパラメータの真値を含む割合**

## 反応率の信頼区間の被覆確率

**演習** : 反応率の推定値に対する95%信頼区間を計算するシミュレーションを1000回行い, 被覆確率を求めよう

サンプルコード シミュレーション設定

```
trials <- 1000 # シミュレーション回数  
n <- 20 # データ数  
p <- 0.4 # 真の反応率  
cover <- numeric(trials)
```

# 反応率の信頼区間の被覆確率

**演習** : 反応率の推定値に対する95%信頼区間を計算するシミュレーションを1000回行い, 被覆確率を求めよう

## サンプルコード

```
set.seed(42)
for(i in 1:trials){
  observation <- rbinom(n, size = 1, prob = p)
  p_est <- sum(observation) / n # 反応率の推定値
  p_est_se <- sqrt(p_est*(1 - p_est)/n) # 標準誤差
  CI_up <- p_est + 1.96*p_est_se # 信頼区間の上限
  CI_low <- p_est - 1.96*p_est_se # 信頼区間の下限
  cover[i] <- as.numeric(CI_low <= p && p <= CI_up)
}
print(paste("被覆確率 : ", mean(cover)))
```

# 反応率の信頼区間の被覆確率

**演習** : 反応率の推定値に対する95%信頼区間を計算するシミュレーションを1000回行い, 被覆確率を求めよう

## サンプルコード

```
set.seed(42)
for(i in 1:trials){
  observation <- rbinom(n, size = 1, prob = p)
  p_est <- sum(observation) / n # 反応率の推定値
  p_est_se <- sqrt(p_est*(1 - p_est)/n) # 標準誤差
  CI_up <- p_est + 1.96*p_est_se # 信頼区間の上限
  CI_low <- p_est - 1.96*p_est_se # 信頼区間の下限
  cover[i] <- as.numeric(CI_low <= p && p <= CI_up)
} # CIがpを含んでいれば1, そうでなければ0を結果として保存
print(paste("被覆確率 : ", mean(cover)))
```

## 反応率の信頼区間の被覆確率

**演習** : 反応率の推定値に対する95%信頼区間を計算するシミュレーションを1000回行い, 被覆確率を求めよう

計算結果

```
print(cover)    CIが真値を含むかどうかの結果
```

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1  
[38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
[75] 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1
```

"被覆確率 : 0.934"

→ 1000回の試行中93.4%の試行で真値を含む結果が得られた



## 推定精度とサンプルサイズの関係

**演習** : 患者数を40人にして, 2000回のシミュレーションで推定値の分布, 分散, 信頼区間がどう変わるかを確認してみよう

### サンプルコード シミュレーション設定

```
trials <- 2000 # シミュレーション回数
n <- 40 # データ数
p <- 0.4 # 真の反応率
p_est_vec <- numeric(trials)
cover <- numeric(trials)
```

# 推定精度とサンプルサイズの関係

**演習** : 患者数を40人にして, 2000回のシミュレーションで推定値の分布, 分散, 信頼区間がどう変わるかを確認してみよう

## サンプルコード

2000回の試行

```
set.seed(42)
for(i in 1:trials){
  observation <- rbinom(n, size = 1, prob = p)
  p_est_vec[i] <- sum(observation) / n # 反応率の推定値
  p_est_se <- sqrt(p_est*(1 - p_est)/n) # 標準誤差
  CI_up <- p_est + 1.96*p_est_se # 信頼区間の上限
  CI_low <- p_est - 1.96*p_est_se # 信頼区間の下限
  cover[i] <- as.numeric(CI_low <= p && p <= CI_up)
}
```

## 推定精度とサンプルサイズの関係

**演習** : 患者数を40人にして, 2000回のシミュレーションで推定値の分布, 分散, 信頼区間がどう変わるかを確認してみよう

### サンプルコード

```
p_est_mean <- mean(p_est_vec) # 2000回の試行の平均反応率
p_est_var <- var(p_est_vec)*(trials-1)/trials # 2000回の試行の分散
CI_low <- p_est_mean - 1.96*sqrt(p_est_var) # 信頼区間の下限
CI_up <- p_est_mean + 1.96*sqrt(p_est_var) # 信頼区間の上限
```

# 推定精度とサンプルサイズの関係

**演習** : 患者数を40人にして, 2000回のシミュレーションで推定値の分布, 分散, 信頼区間がどう変わるかを確認してみよう

## サンプルコード

```
hist(p_est_vec, breaks = 10, col = "skyblue", prob=TRUE,
     main = "2000回の試行で得られた反応率の相対頻度",
     xlab = "反応率の推定値")
print(paste("反応率の推定値 (2000回の平均) :", p_est_mean))
print(paste("分散の理論値 :", p*(1-p)/n))
print(paste("シミュレーション結果から計算した経験分散", p_est_var))
print(paste("反応率の95%信頼区間の下限 :", CI_low))
print(paste("反応率の95%信頼区間の上限 :", CI_up))
print(paste("被覆確率 :", mean(cover)))
```

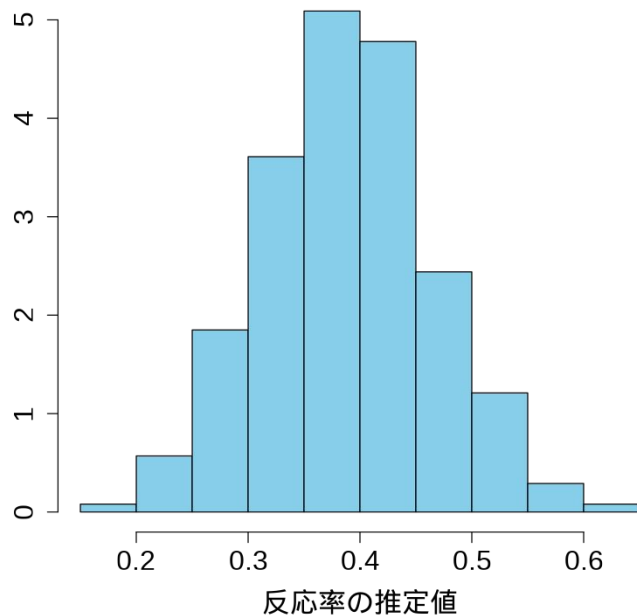
# 推定精度とサンプルサイズの関係

**演習** : 患者数を40人にして, 2000回のシミュレーションで推定値の分布, 分散, 信頼区間がどう変わるかを確認してみよう

2000回の試行で得られた反応率の相対頻度

## シミュレーション結果

- [1] "反応率の推定値 (2000回の平均) : 0.4013875"
- [1] "分散の理論値 : 0.006"
- [1] "シミュレーション結果から計算した経験分散 0.00583838734375"
- [1] "反応率の95%信頼区間の下限 : 0.251625192259534"
- [1] "反応率の95%信頼区間の上限 : 0.551149807740466"
- [1] "被覆確率 : 1"



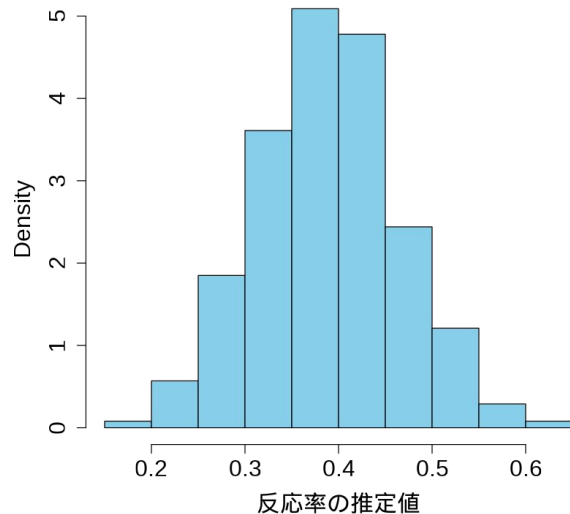
高い精度で正しい反応率が推定できている

# 推定精度とサンプルサイズの関係

**演習** : さらに患者数を増やして, 2000回のシミュレーションで推定値の分布がどう変化していくかみてみよう

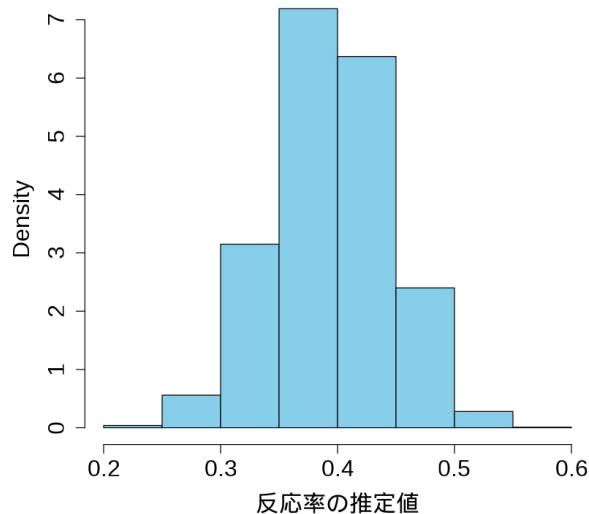
$n=40$

2000回の試行で得られた反応率の相対頻度



$n=100$

2000回の試行で得られた反応率の相対頻度



$n=1000$

2000回の試行で得られた反応率の相対頻度

