可視化と数値実験

グラフ描画とモンテカルロ法

(Press `?` for help, `n` and `p` for next and previous slide)

村田 昇

2019.10.11

描画の基礎

2.1

データの可視化

- データ全体の特徴や傾向を把握するための直感的で効果的な方法
- R言語には極めて多彩な作図機能が用意されている
- 基本となるのは関数 `plot()/curve()`
- 描画関連の関数は色，線種や線の太さ，図中の文字の大きさなどを指定することができる

2.2

演習

- 以下のそれぞれの場合について 03-plot.r を実行しながら確認しよう.

2.3

基本的な描画 (ベクトル)

- ベクトルデータを左から等間隔で描画

```
plot(x, type="p", xlim=NULL, ylim=NULL,
     main=NULL, xlab=NULL, ylab=NULL, ...)
## x: ベクトル
## type: 描画タイプ．既定値は "p"(点プロット)．"l"(折れ線)など指定可
## xlim: x軸の範囲．既定値は自動的に決定
## ylim: y軸の範囲．既定値は自動的に決定
## main: 図のタイトル．既定値はタイトルなし
## xlab: x軸のラベル名．既定値は"Index"
## ylab: y軸のラベル名．既定値はxのオブジェクト名
## ...: 他のオプション．以下に例示．詳細は help(par) を参照
## col: 色の指定．"red"や"blue"など．指定可能な色は colors() を参照
## pch: 点の形．詳細は help(points) を参照
## cex: 文字の大きさ．既定値の何倍にするかを指定
## lty: 線のタイプ．実線・破線などを記号・数字で指定．詳細は help(par) を参照
## lwd: 線の太さ．数字で指定
```

2.4

基本的な描画 (関数)

- 1変数関数の範囲を指定して描画

```
curve(fun,from=NULL,to=NULL, ...)
## fun: 1変数関数
## from: x軸の左端
## to: x軸の右端
## ...: "ベクトルの描画"と同じオプションが利用可能

plot(fun, y=0, to=1, ...)
## curveとほぼ同様
## y: x軸の左端 (from=と書いても良い)

## 別の関数 f を重ね書きする場合
curve(...,add=TRUE, ...)
plot(...,add=TRUE, ...)
```

基本的な描画 (散布図)

- 点 $(x_1, y_1), \ldots, (x_N, y_N)$ を平面上に描画
  - 2つの同じ長さのベクトル $x_1, \ldots, x_N$ と $y_1, \ldots, y_N$ を与える
  - $x_1, \ldots, x_N$ と $y_1, \ldots, y_N$ を持つデータフレームを与える

```
plot(x, y, ...)
## x: 1種類目のデータ x_1,...,x_N
## y: 2種類目のデータ y_1,...,y_N
## ...: "ベクトルの描画"と同じオプションが利用可能

plot(B ~ A, data=x, ...)
## データフレームxの変数A, Bの散布図を作成する
```

2.6

図の保存

- RStudioの機能を使う (少数の場合はこちらが簡便)
  - 右下ペイン “Plots” > “Export”
  - 形式やサイズを指定する
  - クリップボードにコピーもできる
- コマンドで実行する (多数の場合はこちらで処理)
  - 関数 `pdf()`
  - 関数 `png()`
  - 関数 `dev.copy()`

  などを参照

2.7

演習

- 2種類の睡眠薬投与による睡眠時間の増減のデータ `datasets::sleep` において
  - `group` が1のデータの `extra` をx軸
  - `group` が2のデータの `extra` をy軸

  とした散布図を描画せよ(詳細は `help(sleep)`).
- ただし，　点の色を青，　点の形を×，　タイトルを “sleep data”，　x軸のラベルを “group 1”，　y軸のラベルを “group 2” とせよ.

2.8

さまざまなグラフ

3.1

演習

- 以下の各関数について 03-graph.r を実行しながら確認しよう.

3.2

ヒストグラム

- データの値の範囲をいくつかの区間に分割し，各区間に含まれるデータの個数を棒グラフにした図
- 棒グラフの幅が区間, 面積が区間に含まれるデータの個数に比例するようにグラフを作成
- データの分布を可視化するのに有効(どのあたりに値が集中しているか, どの程度値にばらつきがあるかなど)

```
hist(x, breaks="Sturges", freq=NULL)
## x: ベクトル
## breaks: 区間の分割の仕方を指定．数字を指定するとそれに近い個数に等分割
## freq: TRUEを指定すると縦軸はデータ数,
##       FALSEを指定すると縦軸はデータ数/全データ数．既定値はTRUE
## ...: plotで指定できるオプションが利用可能
```

3.3

箱ひげ図

- データの中心，散らばり具合および外れ値を考察するための図 (ヒストグラムの簡易版)
  - 太線で表示された中央値(第2四分位点)
  - 第1四分位点を下端・第3四分位点を上端とする長方形(箱)
  - 中央値から第1四分位点・第3四分位点までの1.5倍以内にあるデータの 最小の値・最大の値を下端・上端とする線(ひげ)
  - ひげの外側のデータは点で表示
- 複数のデータの分布の比較の際に有効

```
boxplot(x, ...)
## x: ベクトルまたはデータフレーム
##       ベクトルに対しては単一の箱ひげ図
##       データフレーム対しては列ごとの箱ひげ図
## ...: plotと同様のオプションを指定可能

boxplot(B ~ A, data=x, ...)
## xの変数Bを変数A(質的変数; 性別, 植物の種類など)で分類する場合
```

3.4

棒グラフ

- 項目ごとの量を並べて表示した図
- 縦にも横にも並べられる

```
barplot(x,width=1,space=NULL,beside=FALSE,
        legend.text=NULL,args.legend=NULL, ...)
## x: ベクトルまたは行列（データフレームは不可）
## width: 棒の幅
## space: 棒グラフ間・変数間のスペース
## legend.text: 凡例
## beside: 複数の変数を縦に並べるか・横に並べるか
## args.legend: legendに渡す引数
## ...: plotで指定できるオプションが利用可能
```

3.5

円グラフ

- 項目ごとの比率を円の分割で表示した図
- 時計回りにも反時計回りにも配置できる

```
pie(x, clockwise=FALSE, ...)
## x: ベクトル
## clockwise: 時計回りに書くか否か
## ...: plotで指定できるオプションが利用可能
```

散布図行列

- 散布図を行列状に並べた図
- データフレームの全ての列の組み合わせの散布図を同時に見ることができる

```
pairs(x, ...)
plot(x,...) # pairsと同じ結果となる
## x: データフレーム

pairs(~ A1 + ... + Ak, data=x, ...)
plot(~ A1 + ... + Ak, data=x, ...)
## 変数A1,...,Akのみ考える場合
```

俯瞰図

- 3次元のグラフを2次元に射影した図

```
persp(x, y, z, theta=0, phi=15, expand=1)
## x,y,z: x,y,z座標
##        (zは行列で，z[i,j]は点(x[i],y[j])に対応する値を与える)
## theta,phi: 俯瞰の方向を指定する極座標
## expand: z軸の拡大度
## ...: plotで指定できるオプションが利用可能

## 多様な3次元のグラフのためのパッケージがある．
## 以下はscatterplot3dパッケージの例
scatterplot3d(x, color, angle=40)
## x: x,y,z座標を指定するデータフレーム
##    (perspのように直接指定することも可能)
## color: 色を指定(colではないので注意)．既定値は黒
## angle: x軸とy軸の間の角度
## ...: plotで指定できるオプションが利用可能
```

グラフィクス環境の設定

- グラフィクス関数には様々なオプションがある
- 共通の環境設定のためには関数 `par()` を用いる
  - 複数の図の配置: `mrow`, `mcol`
  - 余白の設定: `margin`
  - 日本語フォントの設定: `family`
  - 他多数 (`?par` を参照)
  - より進んだグラフィクスの使い方の例は `demo("graphics")`, `example(関数名)` を参照

3.9

演習

- 適当なデータに対してグラフの作成を行ってみよう
  - 東京都の気候データ (tokyo-weather.csv)
  - R言語に用意されているデータ (関数 `data()` で一覧表示)

疑似乱数

疑似乱数とは

- コンピューターで生成された数列のこと
- 完全にランダムに数字を発生されることは不可能
- Rの既定値は “Mersenne-Twister法” (`?Random` 参照)
- 数値シミュレーションにおいて再現性が要請される場合には, 乱数の “シード値” を指定して再現性を担保 (関数 `set.seed()`)

基本的な乱数

- **ランダムサンプリング**: 与えられた集合の要素を無作為抽出することで発生する乱数
- **二項乱数**: 「確率 p で表がでるコインを n 回投げた際の表が出る回数」に対応する乱数
- **一様乱数**: 決まった区間 (a, b) からランダムに発生する乱数
- **正規乱数**: 平均 μ, 分散 σ^2 の正規分布に従う乱数

乱数を生成する関数

- 関数 `sample()`: ランダムサンプリング
- 関数 `rbinom()`: 二項乱数
- 関数 `runif()`: 一様乱数
- 関数 `rnorm()`: 正規乱数

演習

- 03-random.r を確認してみよう

4.5

モンテカルロ法

5.1

モンテカルロ法とは

- 乱数を使った統計実験
- 計算機上でランダムネスを実現 (擬似乱数)
- ランダムネスから導かれる種々の数学的結果を観察

5.2

推定量の分布

- **推定量**: 確率分布の特性値を推測する計算方法
- **推定値**: 観測データから計算した値
- 推定値は観測データに依存して異なる (ばらつく)
- 推定量の分布を求める
  - 理論的な解答: 確率の理論を用いて厳密に求める
  - 数値的な解答: 疑似乱数を用いた数値実験で求める

標本平均の分布

- 標本平均: 平均値の典型的な推定量

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

- データ数が十分大きいとき標本平均の理論的な分布は，観測データの分布(正規分布とは限らない)の平均 μ と分散 σ^2 を用いて与えられる.
(次に述べる **中心極限定理** による)
- 標本平均は平均 μ，分散 σ^2/n の正規分布に従う.

中心極限定理

- $X_1, X_2, \ldots$ を独立同分布な確率変数列とし，その平均を μ，標準偏差を σ とする．このとき，すべての実数 $a < b$ に対して

$$P\left(a \leq \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \leq b\right) \to \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx \quad (n$$

が成り立つ．

5.5

演習

- 03-meandist.r を確認してみよう

5.6

確率的な数値実験

- 例: コイン投げの賭け
  AとBの二人で交互にコインを投げる． 最初に表が出た方を勝ちとするとき， AとBそれぞれの勝率はいくつとなるか？
- コイン投げは `sample()`, `rbinom()` などで模擬できる

5.7

演習

- 03-cointoss.r を確認してみよう

5.8

演習

- 以下の簡単な双六ゲームの実験を行ってみよう
  - ゴールまでの升目は100
  - さいころを振り出た目の数だけ進む
  - ゴールに辿り着くまで繰り返す
  - さいころを振る回数の分布は?

5.9

演習

- 確率的な事象を想定して実験を行ってみよう
  - Buffon の針
  - Monty Hall 問題
  - St Petersburg のパラドックス
  - 秘書問題 (最適停止問題)
- (講義資料に実装例があります)