# データの可視化

第 5 講 - 様々なグラフの描画

村田 昇

## 講義の内容

- 可視化の重要性
- 基本的な描画
- 分布の視覚化
- 比率の視覚化
- 多次元データの視覚化

## 可視化の重要性

### 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
  - **base R** : `package::graphics` (標準で読み込まれる)
  - **tidyverse** : `package::ggplot2`
- 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

### サンプルデータの説明

- `jpdata[1-3].csv` (再掲)
  - https://www.e-stat.go.jp (統計局)
    - 地域から探す / 全県を選択 / 項目を選択してダウンロード
    - 日本語が扱えることを想定して日本語を含んでいる
    - 英語のために -en を用意
  - データファイル (文字コード : utf8)
    - jpdata1.csv : 県別の対象データ
    - jpdata2.csv : 対象データの内容説明
    - jpdata3.csv : 県と地域の対応関係
  - 作業ディレクトリの data 内に置いて読み込む場合

```
jp_data <- read_csv(file = "data/jpdata1.csv")
jp_item <- read_csv(file = "data/jpdata2.csv")
jp_area <- read_csv(file = "data/jpdata3.csv")
```

    - 変数名は自由に付けてよい

- `tokyo_weather.csv` (`tokyo.zip` の中)

  - https://www.jma.go.jp (気象庁)
    * 各種データ・資料 / 過去の地点気象データ・ダウンロード
    * 地点 / 項目 / 期間を選択してダウンロード
    * ダウンロードしたものを必要事項のみ残して整理
  - データ項目平均気温 (℃), 降水量の合計 (mm), 合計全天日射量 (MJ/㎡), 降雪量合計 (cm), 最多風向 (16 方位), 平均風速 (m/s), 平均現地気圧 (hPa), 平均湿度 (%), 平均雲量 (10 分比), 天気概況 (昼：06 時〜18 時), 天気概況 (夜：18 時〜翌日 06 時)
  - 作業ディレクトリの data 内に置いて読み込む場合

```
tw_data <- read_csv(file = "data/tokyo_weather.csv")
```

- tokyo_covid19_2021.csv (tokyo.zip の中)
  - https://stopcovid19.metro.tokyo.lg.jp (東京都)
  - データ項目陽性者数, 総検査実施件数, 発熱等相談件数
  - 作業ディレクトリの data 内に置いて読み込む場合

```
tc_data <- read_csv(file="data/tokyo_covid19_2021.csv")
```

# 描画の基礎

## 描画の初期化

- package::ggplot2 ではさまざまな作図関数を追加しながら描画する

```
初期化のための関数 +
  作図のための関数 + ... +
  装飾のための関数 + ... # 関数が生成するオブジェクトに変更分を随時追加する
```

- 関数 ggplot2::ggplot() : 初期化

```
ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())
#' data: データフレーム
#' mapping: 描画の基本となる"審美的マップ"(xy 軸, 色, 形, 塗り潰しなど) の設定
#' environment: 互換性のための変数 (廃止)
#' 詳細は '?ggplot2::ggplot' を参照
```

## 基本的な描画 (折線グラフ)

- 関数 ggplot2::geom_line() : 線の描画

```
geom_line(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE, orientation = NA, show.legend = NA, inherit.aes = TRUE
)
#' mapping: "審美的"マップの設定
#' data: データフレーム
#' stat: 統計的な処理の指定
#' position: 描画位置の調整
#' ...: その他の描画オプション
#' na.rm: NA(欠損値) の削除 (既定値は削除しない)
#' show.legend: 凡例の表示 (既定値は表示)
#' 詳細は '?ggplot2::geom_line' を参照
```

- 行政検査と医療機関の検査件数の推移

```
pcr_data |> # パイプ演算子でデータフレームを関数 ggplot2::ggplot() に渡す
  ggplot(aes(x = date)) + # date を x 軸に指定
  geom_line(aes(y = ai), colour = "blue") + # 行政検査を青
  geom_line(aes(y = mi), colour = "red") +  # 医療機関を赤
  labs(y = "number of tests") # y 軸のラベルを変更
```

□□□□□□ 'pcr_data' □□□□□□

- 全ての機関の検査件数の推移

```
pcr_data |> select(!c(sub,total)) |> # 集計値を除く
  pivot_longer(!date, names_to = "organ", values_to = "nums") |>
  ggplot(aes(x = date, y = nums, colour = organ)) + geom_line() +
  labs(title = "PCR Tests in Various Organizatios",
       x = "Date", y = "Number of Tests") # xy 軸のラベルを変更
```

□□□□□□ 'pcr_data' □□□□□□

- 別の形式での描画

```
pcr_data |> select(!c(sub,total)) |>
  pivot_longer(!date, names_to = "organ", values_to = "nums") |>
  ggplot(aes(x = date, y = nums, colour = organ)) +
  labs(title = "PCR Tests in Various Organizatios", x = "Date", y = "Number of Tests") +
  geom_line(show.legend  =  FALSE) + # 凡例を消す
  facet_grid(vars(organ)) # "organ" ごとに異なる図を並べる
```

□□□□□□ 'pcr_data' □□□□□□

## 基本的な描画 (散布図)

- 関数 ggplot2::geom_point() : 点の描画

```
geom_point(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE
)
#' mapping: 審美的マップの設定
#' data: データフレーム
#' stat: 統計的な処理の指定
#' position: 描画位置の調整
#' ...: その他の描画オプション
#' na.rm: NA(欠損値) の削除 (既定値は削除しない)
#' show.legend: 凡例の表示 (既定値は表示)
#' 詳細は '?ggplot2::geom_point' を参照
```

- 国立感染症研究所と医療機関の検査件数の関係

```
if(Sys.info()["sysname"] == "Darwin") { # MacOS か調べて日本語フォントを指定
  theme_update(text = element_text(family = "HiraginoSans-W4"))}
pcr_data |>
  ggplot(aes(x = niid, y = mi)) + # x 軸を niid, y 軸を mi に設定
  geom_point(colour = "blue", shape = 19) + # 色と形を指定 (点の形は '?points' を参照)
  labs(x = pcr_colnames["niid"], y = pcr_colnames["mi"]) # 軸の名前を指定
```

□□□□□□ 'pcr_data' □□□□□□

- 各軸を対数表示

```
pcr_data |>
  ggplot(aes(x = niid, y = mi)) +
  geom_point(colour = "blue", shape = 19) +
  scale_x_log10() + scale_y_log10() + # 各軸を対数で表示
  labs(x = pcr_colnames["niid"], y = pcr_colnames["mi"])
```

□□□□□□ 'pcr_data' □□□□□□

## 基本的な描画 (散布図行列)

- 散布図行列は複数の散布図を行列状に配置したもの
- 関数 GGally::ggpairs() : 散布図行列の描画

```
#' 必要であれば 'install.packages("GGally")' を実行
library(GGally) # パッケージのロード
ggpairs(
  data, mapping = NULL,
  columns = 1:ncol(data),
  upper = list(continuous = "cor", combo = "box_no_facet", discrete = "count", na = "na"),
  lower = list(continuous = "points", combo = "facethist", discrete = "facetbar", na = "na"),
  diag = list(continuous = "densityDiag", discrete = "barDiag", na = "naDiag"),
  ...,
  axisLabels = c("show", "internal", "none"),
  columnLabels = colnames(data[columns]),
  legend = NULL
)
#' columns: 表示するデータフレームの列を指定
#' upper/lower/diag: 行列の上三角・下三角・対角の表示内容を設定
#' axisLabels: 各グラフの軸名の扱い方を指定
#' columnLabels: 表示する列のラベルを設定 (既定値はデータフレームの列名)
#' legend: 凡例の設定 (どの成分を使うか指定)
#' 詳細は '?GGally::ggpairs' を参照
```

- 各検査機関での検査件数の関係を視覚化

```
pcr_data |>
  select(!c(date,sub,total)) |> # 日付と集計値を除いて必要なデータフレームに整形
  ggpairs() # 標準の散布図行列
```

□□□□□□ 'pcr_data' □□□□□□

- 日付の情報を付加

```
pcr_data |> select(!c(sub,total)) |> # 日付から四半期の因子を作成
  mutate(quarter = as_factor(quarter(date, with_year = TRUE))) |>
  ggpairs(columns = 2:8, columnLabels = pcr_colnames[-c(1,8,10)], axisLabels = "none",
          aes(colour = quarter), legend = c(2,1), # 四半期ごとに色づけて (1,1) の凡例を使用
          upper = "blank", diag = list(continuous = "barDiag")) +
  theme(legend.position = "top") # 凡例を上に表示
```

□□□□□□ 'pcr_data' □□□□□□

## 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** タブから **Export** をクリック
  - 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

- 関数 ggsave() : 図の保存

```
ggsave(
  filename,
  plot = last_plot(),
  device = NULL,
  path = NULL, scale = 1, width = NA, height = NA,
  units = c("in", "cm", "mm", "px"), dpi = 300, limitsize = TRUE, bg = NULL,
  ...
)
#' filename: ファイル名
#' plot: 保存する描画オブジェクト
#' device: 保存する形式 ("pdf","jpeg","png"など)
#' 詳細は"?ggplot2::ggsave"を参照
```

### 練習問題

- pcr_case_daily.csv を用いて以下の描画を行いなさい
  - 検疫所 (b)，地方衛生研究所. 保健所 (c)，民間検査会社 (d) における検査件数の推移
  - 民間検査会社 (d)，大学等 (e)，医療機関 (f) での検査件数の関係 (散布図)

## さまざまなグラフ

### 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
  - データ分布の可視化に有効 (値の集中とばらつきを調べる)

```
geom_histogram(
  mapping = NULL, data = NULL, stat = "bin", position = "stack",
  ...,
  binwidth = NULL,
  bins = NULL,
  na.rm = FALSE, orientation = NA, show.legend = NA, inherit.aes = TRUE
)
#' binwidth: ヒストグラムのビンの幅を指定
#' bins: ヒストグラムのビンの数を指定
#' 詳細は '?ggplot2::geom_histogram' を参照
```

- 行政検査での検査件数の分布

```
#' 行政検査 (ai) での検査件数の分布
pcr_data |>
  ggplot(aes(x = ai)) + # 分布を描画する列を指定
  geom_histogram(bins = 30, fill = "lightblue", colour = "blue") +
  labs(x = pcr_colnames["ai"], y = "頻度", title = "検査件数のヒストグラム")
```

□□□□□□ 'pcr_data' □□□□□□

## 箱ひげ図

- データ散らばり具合を考察するための図
  - 長方形の辺は四分位点 (下端が第 1，中央が第 2，上端が第 3)
  - 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)
  - ひげの外側の点は外れ値

```
geom_boxplot(
  mapping = NULL, data = NULL, stat = "boxplot", position = "dodge2",
  ...,
  outlier.colour = NULL, outlier.color = NULL, outlier.fill = NULL,
  outlier.shape = 19, outlier.size = 1.5, outlier.stroke = 0.5, outlier.alpha = NULL,
  notch = FALSE, notchwidth = 0.5, varwidth = FALSE,
  na.rm = FALSE, orientation = NA, show.legend = NA, inherit.aes = TRUE
)
#' ourlier.*: 外れ値の描画方法の指定
#' notch*: ボックスの切れ込みの設定
#' varwidth: ボックスの幅でデータ数を表示
#' 詳細は '?ggplot2::geom_boxplot' を参照
```

- 月ごとの大学等での検査件数の分布 (分位点)

```
#' 大学等 (univ) での検査件数の分布 (2021 年分)
pcr_data |>
  filter(year(date) == 2021) |> # 2021 年を抽出
  mutate(date = as_factor(month(date))) |> # 月を因子化する
  ggplot(aes(x = date, y = univ)) + # 月毎に集計する
  geom_boxplot(fill = "orange") + # 塗り潰しの色を指定
  labs(title = "月ごとの検査件数 (2021 年)", x = "月", y = pcr_colnames["univ"])
```

□□□□□□ 'pcr_data' □□□□□□

## 棒グラフ

- 項目ごとの量を並べて表示した図
  - 並べ方はいくつか用意されている
    - 積み上げ (stack)
    - 横並び (dodge)
    - 比率の表示 (fill)

```
geom_bar(
  mapping = NULL, data = NULL, stat = "count", position = "stack",
  ...,
  just = 0.5,
  width = NULL,
  na.rm = FALSE, orientation = NA, show.legend = NA, inherit.aes = TRUE
)
#' just: 目盛と棒の位置の調整 (既定値は真中)
#' width: 棒の幅の調整 (既定値は目盛の間隔の 90%)
#' 詳細は '?ggplot2::geom_bar' を参照
```

- 機関ごとの月の検査件数の推移

```
#' 機関ごとの月の検査件数の推移 (2021 年分)
pcr_data |>
  filter(year(date) == 2021) |>
  mutate(month = as_factor(month(date))) |> # 月を作成
  select(!c(date,sub,total)) |> # 機関に限定
  group_by(month) |> # 月でグループ化
  summarize(across(everything(), sum)) |> # 全て (月以外) を集計
  pivot_longer(!month, names_to = "organ", values_to = "nums",
               names_transform = list(organ = as_factor)) |>
  ## 最後のオプションは organ 列のラベルを出てきた順で因子化して元の列の並びにしている
  ggplot(aes(x = organ, y = nums, fill = month)) +
  geom_bar(stat = "identity", position = "dodge", na.rm = TRUE) +
  theme(legend.position = "top") + guides(fill = guide_legend(nrow = 1))
```

□□□□□□ 'pcr_data' □□□□□□

### 練習問題

- 適当なデータに対してグラフの作成を行ってみよう
  - PCR 検査件数データ (pcr_case_daily.csv)
  - 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)

## 演習

### 練習問題

- jpdata1/3.csv (前回配布のデータ) を用いて以下の問に答えよ.
  - 婚姻・離婚率の散布図を描け.
  - 地方別に異なる点の形状を用いた散布図を描け.
  - それ以外にも様々な散布図を描画してみよう.
  - (参考) 読み込み方

```
## CSV ファイルは作業ディレクトリの下の data サブディレクトリにあるとする
jp_data <- read.csv(file="data/jpdata1.csv", fileEncoding="utf8", row.names=1)
jp_area <- read.csv(file="data/jpdata3.csv", fileEncoding="utf8")
```

## 演習

### 練習問題

- 配布したサンプルデータ
  - jpdata1.csv
  - tokyo_weather.csv
  - covid19_tokyo.csv

  - `covid19_tokyo_patients.csv`

を用いて以下の問いに答えよ．

  - 3 次元の散布図を作成せよ．
  - 凡例を加えたグラフを作成せよ．

## 次回の予定

- 計算機による数値実験
- 乱数とは
- 乱数を用いた数値実験