Microsoft Excel で作られたデータの取り扱い

演習の目的

- R にデータを正しく読み込むための前処理の基本
- よくある Excel や CSV の問題点とその対処法
- 適切なデータセットとは

演習の準備

・Rstudio の作業ディレクトリの指定

Rstudio で既にプロジェクトを作成している場合は
Console の上部分 R の version の横に表示されている
以下のコマンドで Console 上でも確認、変更できる

```
# 現在の作業ディレクトリの確認
getwd()
# 作業ディレクトリの指定
# setwd("指定したいディレクトリへのpath")
```

・作業ディレクトリへのデータのダウンロード

Excel ファイルを読み込んでみよう

- パッケージ”readxl” を使用すると直接 Excel ファイルを読み込める

```
install.packages("readxl")
library(readxl)
df_example1 <- read_excel("example1.xlsx")
as.data.frame(df_example1)
```

問題点

読み込まれたデータと元データの比較

	A	B	C	E	F
1	日付	陽性数	陰性数	陽性率(%)	
2	2024/4/1	18	102	15.0%	
3	2025/4/2	21	114	15.6%	
4	2025/04/03	16	112	12.5%	
5	2025/04/04	22	118	15.7%	
6	2025/04/05	19	113	14.4%	
7	4月6日	10	90	10.0%	
8					

Excel 上の表示

```
> example1
        日付 陽性数 陰性数 検査数 陽性率 (%)
1      45383     18    102    120     15.00000
2      45749     21    114    135     15.55556
3 2025/04/03     16    112    128     12.50000
4 2025/04/04     22    118    140     15.71429
5 2025/04/05     19    113    132     14.39394
6      45753     10     90    100      0.10000
```

R で読み込んだ表示

- 列の数が異なる
- 日付が正しく読み込めない
- 陽性率の値が異なる

プレーンテキスト形式の推奨

- 列の数が異なる
  → Excel 上に非表示の列が存在
- 日付が正しく読み込めない
  → シリアル値と文字列の日付が混在
- 陽性率の値が異なる
  → 0–1 の数値にパーセント表示を適用したセルと
  0–100 の値にパーセント記号をつけたセルが混在

データ解析には、Excel ではなく、書式情報を持たない csv や txt などのプレーンテキスト形式が推奨される

文字コードに関する注意点

- Mac や RStudio：UTF-8 を前提とする
- Windows：Shift-JIS（CP932）で保存されることがある

→ 文字コードの違いにより文字化けのリスクがあるため
UTF-8 形式で保存・読み込みするのが安全

文字化けが起きたときの対処

・fileEncoding を指定して読み込む

```
# RStudio
# 指定なしだとError
df_example1 <- read.csv("example1_cp932.csv")
# fileEncoding の指定
df_example1 <- read.csv("example1_cp932.csv",
    fileEncoding="CP932")
```

複数行ヘッダーの問題

- 上部にタイトルや単位行があり、実際の列名は 2 行目以降
- read.csv では skip 引数で対応可能
- csv 化する際に削除でも OK

```
df_example2 <- read.csv("example2.csv", skip = 2)
```

出典：厚生労働省「新型コロナウイルス感染症に関するオープンデータ」（https://www.mhlw.go.jp/stf/covid-19/open-data.html）をもとに加工

R の列名に関する制限と注意点

- 数字で始まる列名は X が自動で付加される（例：1 月 → X1 月）
- 空白や記号（例：( ) - / @）は . に変換される（例：score 1 → score.1）
- 予約語と同じ列名にはピリオドが付加される（例：if → if.）
- 重複する列名は自動で番号が付く（例：score, score → score, score.1）
- 空文字・NA は X, NA. に置換されることがある

→ 半角英数字＋アンダースコア（_）で構成された列名が推奨

良いデータセットとは？

- 各列が「1 つの変数」
- 各行が「1 つの観測単位（レコード）」
- 不要な装飾・合計行が含まれない
- 欠損値は NA などで明示的に表現されている
- 列名は扱いやすく（英数字＋ _ など）

データ解析しやすい構造にすることが重要
（整理された構造 $\rightarrow$ 処理・再利用が容易）

Wide format（横持ち）

Wide format の特徴

- 1 人や 1 施設が 1 行にまとまっており、列が多くなる形式
- 入力、確認しやすい

例 1：新規感染者数データ

Prefecture	2020/07/01	2020/07/02	2020/07/03
Tokyo	67	107	124
Osaka	10	8	11
Hokkaido	4	8	0

例 2：バイタルサイン（体温・血圧、仮想データ）

Patient ID	Temp_AM	Temp_PM	BP_AM	BP_PM
P001	36.5	36.8	120	115
P002	37.0	37.2	130	125

Long format（縦持ち）

Long format の特徴

- 1 人複数行のデータ
- 個人ごとに測定時点、回数が違っても使用できる
- 分析や可視化、整形に適した構造

例 1：新規感染者数

Prefecture	Date	New cases
Tokyo	2020/07/01	67
Tokyo	2020/07/02	107
Tokyo	2020/07/03	124
Osaka	2020/07/01	10
Osaka	2020/07/02	8
Osaka	2020/07/03	11
Hokkaido	2020/07/01	4
Hokkaido	2020/07/02	8
Hokkaido	2020/07/03	0

例 2：バイタルサイン（体温・血圧）

Patient ID	Time	Item	Value
P001	AM	Temp	36.5
P001	PM	Temp	36.8
P001	AM	BP	120
P001	PM	BP	115
P002	AM	Temp	37.0
P002	PM	Temp	37.2
P002	AM	BP	130
P002	PM	BP	125

R で読み込んだデータの確認

- class( )：オブジェクト全体や列の型を確認
- dim( )：行数と列数（dimensions）の確認
- str( )：列名・型・一部の値を確認
- summary( )：数値の要約統計量や欠損の有無を確認
- head( )：先頭 6 行を表示（中身の様子をざっくり確認）

```
# CSVファイルの読み込み
df_example3 <- read.csv("example3.csv")
# データの型を確認
class(df_example3)
# データの大きさを確認
dim(df_example3)
# 構造の確認（列名・型・一部の値）
str(df_example3)
# 要約統計量（平均・最小・最大など）と欠損の有無を確認
summary(df_example3)
# 先頭6行を表示
head(df_example3)
```

カテゴリ変数の扱い

- 医療データには「性別」「重症度」などのカテゴリ変数が多く含まれる
- R では「カテゴリ変数（名義尺度・順序尺度）」は factor 型で扱われる
- 分類、グループ化、統計解析で重要

```
# 文字列の列をfactor型で読み込み
df_example3 <- read.csv("example3.csv",
    stringsAsFactors = TRUE)
# summary()関数で違いを確認
summary(df_example3)
```

R における欠損値の扱い

- R では、値が存在しない（欠損している）ことを NA として表現
- NA は、数値型・文字型・論理型などすべての型で扱われる特別な値
- 計算や論理比較に含まれると、結果も NA になる
- 欠損の判定には is.na() を使用

```
# 欠損のある列を確認
summary(df_example3)
# (temperatureに欠測あり, statusは一部の欠損がカテゴリとして
    認識されている)
# 欠損かどうかの判定
is.na(df_example3)
# 欠損の個数を集計
sum(is.na(df_example3))
```

演習問題

1. 実行中の Rstudio のプロジェクトの作業ディレクトリを特定してください
2. `exercise1.xlsx` を Excel で開き csv ファイルで保存してください
    - ヘッダーフッターの削除
    - 列名の変更
    - 不要な列の削除
    - 保存ファイルは文字コード utf-8 の csv ファイル
    - 保存先は特定した作業ディレクトリ
3. 修正した csv ファイルを R 上で読み込んでください
4. R で読み込んだデータを確認してください
    - 関数 class() を用いる
    - 関数 summary() を用いる

参考: 欠損値の処理

・ifelse() を使用した NA への変換

```
# データフレームの列の抽出
df_example3$status
df_example3[,6]
# Unknownを欠損値扱いとする変換
# 誤：数値になってしまう
ifelse(df_example3$status == "Unknown", NA,
    df_example3$status)
# 正：文字列として処理
df_example3$status <- ifelse(as.character(
    df_example3$status) == "Unknown", NA, as.character
    (df_example3$status))
df_example3$status <- factor(df_example3$status) # 必
    要なら再変換
summary(df_example3)
```

dplyr パッケージの na_if() 関数も有用

参考: 演習問題（発展的内容, 資料のみ）

1. 読み込んだ exercise1 のデータの各列のデータ型を確認してください
2. 文字型として読み込まれている列を数値型にしてください
3. データを long format にしてください
4. 各県での入院を要する症例数と退院・療養解除した症例数の時系列の変動を示す図を ggplot を用いて作成してください

参考: 演習問題（発展的内容, 資料のみ）

例: