オープンソースソフトウェアの活用事例紹介

日本製薬工業協会　医薬品評価委員会　データサイエンス部会

2023 年度 タスクフォース 5

2024 年 6 月

目次

略語一覧表

略語	略していない表現（英語）	略していない表現（日本語）
OSS	Open Source Software	オープンソースソフトウェア
CRAN	Comprehensive R Archive Network	包括的 R アーカイブネットワーク
API	Application Programming Interface	ソフトウェアやプログラム、Web サービスを相互につなぐインタフェース

1. はじめに

日本製薬工業協会 医薬品評価委員会のデータサイエンス部会 2022 年度タスクフォース 5 で実施された OSS 利用状況アンケートの報告書[1)]では「R/Python 等のオープンソースの利用に関して今後製薬協タスクフォースに期待することを教えてください」の質問に対する回答として、承認申請利用（規制当局の OSS 受入れ可否、承認申請時の申請者で担保すべき品質、管理体制）に次ぎ、OSS 利用事例紹介（OSS で何ができるかの活用事例紹介や、サンプルプログラムの公開）の回答が多かった。そこで、2023 年度のタスクフォース 5 では、タスクフォースメンバー所属会社での Python と R を使った事例を収集し、これらを文書としてまとめてサンプルプログラムと共に公開することにした。

本文書は Python 活用事例紹介と R 活用事例紹介の 2 つのパート分けて事例紹介しており、それぞれの事例に対して機能概要、処理概要、プログラムの実行環境について記載した。本文書で提示している事例には、マーケティング関連のものや、機能の重複しているものが存在しているが、可能な限り多くの事例を提供する事を目的としているため、あえて提示していることをご理解いただきたい。

1) OSS 利用状況アンケート報告書
https://www.jpma.or.jp/information/evaluation/results/allotment/g75una0000001axl-att/DS_202204_oss_questionaire.pdf

2. 事例紹介

2.1 Python 活用事例紹介

2.1.1 ClinicalTrials.gov から関心のある疾患の試験情報を取得する

1） 機能概要

ClinicalTrials.gov[1)]の API[2)]を使って、関心のある疾患の試験 ID（NCT ID)、試験タイトル、試験概要情報を取得し Excel ファイルに出力する。

2） 処理概要

ClinicalTrial.gov の API に関心のある疾患と、必要情報が格納されているフィールド名称（試験 ID: NCTId, 試験タイトル: BriefTitle, 試験概要: BriefSummary)、出力形式（csv 形式か json 形式)、取得する試験数（10 件）を指定する。この時、API に渡す URL の構成は以下のようになる。

```
https://clinicaltrials.gov/api/v2/studies?query.cond=osteogenesis+imperfecta&fields=NCTId%2CBriefTitle%2CBriefSummary&pageSize=10
```

緑色のパラメータは ClinicalTrials.gov の API の URL となる。本 URL に各種パラメータを渡すため「?」以降に必要パラメータを設定する。黄色のパラメータ（query.cond=）は関心のキーワードを渡すパラメータで、本例では関心のある疾患として「osteogenesis imperfecta」を指定している（薬剤名の指定も可能）。水色のパラメータ（fields=）は必要情報が格納されているフィールド名称[3]を指定している。この時、複数のフィールドを取得したい場合は、対象フィールドをカンマ（，）で区切って指定する。赤色のパラメータ(pageSize=)では、検索結果の出力数を 10 に指定している。これらのパラメータを「&」でつなげて API としてリクエストを送信する URL 構成する（設定可能なパラメータは他にも多く存在しており、詳細は ClinicalTrials.gov の API サイト[2]で確認できる）。

Python の requests パッケージ 4)の get 関数で API リクエストを送信し、検索結果を json 形式で取得できる。json 形式の検索結果は、json パッケージ[5]の loads 関数で辞書型データに変換した後、目的のフィールド情報の値を取得して Excel 出力する。フィールドは決められた階層構造（study data structure[3]）をとっており、本情報を元に、辞書データの key 値を設定することで情報を取得している。

実際のプログラムコードを以下に示す。

```
for s in studies:
  # NCTId
  id = s['protocolSection']['identificationModule']['nctId']

  # BriefTitle                                                    ①
  btitle = s['protocolSection']['identificationModule']['briefTitle']

  # BriefSummary                                                  ②
  bsummary = s['protocolSection']['descriptionModule']['briefSummary']

  # 試験情報を配列に格納する
  sheet.append([id, btitle, bsummary])
```

①部分で BriefTitl を取得する処理になるが、ここで指定する処理になるが、ここで指定する辞書データの key 値は図 2.1.1-1 Study data structure[2]で示した①の階層情報を元に取得している。②の BriefSummary も同様に「図 2-1-1-1 Study data structure」[2]の階層情報を元に取得している。

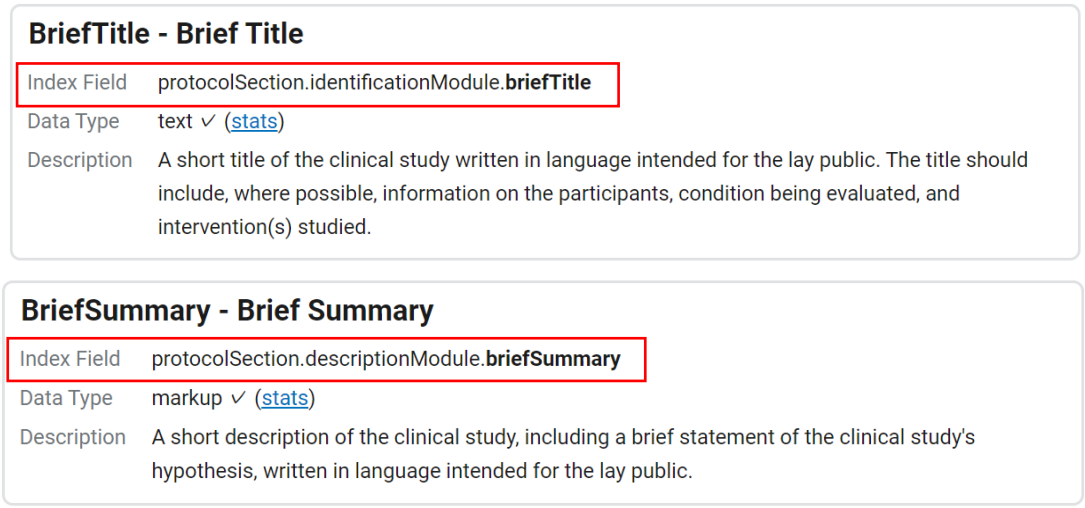


図 2.1.1-1 Study data structure

また、検索結果が 10 件を上回る場合、次の 10 件のデータの検索結果を取得するためのトークンが json 形式のキー（nextPageToken）に設定される。このトークンを検索 URL のパラメータ（pageToken）に追加することで次の 10 件のデータにアクセスできるようになる。

https://clinicaltrials.gov/api/v2/studies?query.cond=osteogenesis+imperfecta&fields=NCTId%2CBriefTitle%2CBriefSummary&pageSize=10&pageToken=NF0g5JeDlvMrwA

上記 URL での検索結果に次の 10 件の検索結果がある場合、同様に次の 10 件のデータの検索結果を取得するためのトークンが json 形式のキー（nextPageToken）に設定される。トークン情報が設定されたキー（nextPageToken）が設定されなくなるまで、while ループ処理で検索結果を取得している。

実際のプログラムコードを以下に示す。

```
# 次ページ情報が存在する場合、情報を取得し続ける
while NEXTPAGE in dictCTInfo:

  # next page token の取得
  token = dictCTInfo['nextPageToken']

  # 次ページの情報を取得
  new_API_URL = API_URL + '&' + NEXTPAGETOKEN + '=' + token
  print('NOTE: Next page found ====>')
  rint('ClinicalTrial.gov API URL:')
  print(new_API_URL)
  print('-----------------------------------------------')
```

```
# API URL に HPPT リクエストを送信し、実行結果（json 形式）を辞書型で取得する
dictCTInfo = http_request(new_API_URL)

# 検索結果の試験情報を配列に追加する
studies.extend(dictCTInfo['studies'])
```

while ループ処理を抜ける時には、すべての試験の情報が辞書形式となって配列に格納されることになる。

3) 実行環境

Objects	Version
Python	3.10.9
pandas	1.5.3
requests	2.28.1
openpyxl	3.0.10
json	2.0.9

参考文献

1) ClinicalTrials.gov
https://clinicaltrials.gov/
臨床試験情報（試験概要や試験薬、エンドポイント、試験結果等）が格納されているデータベース

2) API
https://clinicaltrials.gov/data-api/api
ClinicalTrial.gov で公開しているデータベース検索用の API サイト

3) フィールド名称
ClinicalTrial.gov の API 検索結果は csv か json 形式で出力される。json 形式の場合に、各種情報が格納されている key 情報が決められており、それらの一覧は以下の URL から参照可能。
https://clinicaltrials.gov/data-api/about-api/study-data-structure

4) requests パッケージ
HTTP リクエストの送信と結果の取得を制御する Python パッケージ。
https://pypi.org/project/requests/2.28.1/

5) json パッケージ
json 形式の作成や読み込みを目的とした Python パッケージ。Python の標準ライブラリに含まれている。

2.1.2　Annotated CRF（aCRF）の注釈を別の blank CRF へコピーする

1)　機能概要

aCRF にアノテーションされている SDTM のドメインや変数等のテキストボックスを別の blank CRF の同じしおりを持つページにコピーする。(CRF のバージョンアップの際にアノテーションを載せ替える作業時に利用する想定)

2)　処理概要

ソースの PDF とコピー先の PDF からしおりの情報とページ情報を辞書型で取得する。また、ソースの PDF 内のテキストボックスの情報を取得し、リスト化する。その後、取得したしおり情報及びページ情報に従ってテキストボックスを pdfrw.PdfWriter でコピー先の PDF に書き込みする。
本機能は 2 つ関数で実現している。

一つ目の関数である flatten_outlines(outlines)は入れ子になった多次元配列を 1 次元配列（リスト）に変換する関数である。

```
def flatten_outlines(outlines):

  flattened = []  ①

  for outline in outlines:
  if isinstance(outline, list):
    flattened.extend(outline)
  else:
    flattened.append(outline)

  return flattened  ③
```

②

引数 outlines は、入力 PDF のしおり情報が格納された多次元配列をパラメータとして指定している。
処理は以下の流れとなる。

① 空のリスト flattened を作成する。
② 引数 outlines の各要素に対して、要素がリストである場合、リストの各要素を flattened に追加する。要素がリストでない場合、その要素を直接 flattened に追加する。
③ 一次元配列に変換された flattened を戻り値とする

二つ目の関数である copy_annotations(src_pdf, dest_pdf, output_pdf)は、PDF ファイル間でアノテーションをコピーする関数である。

```
def copy_annotations(src_pdf, dest_pdf, output_pdf):
    src_reader = PyPDF2.PdfFileReader(src_pdf)
    dest_reader = PyPDF2.PdfFileReader(dest_pdf)
    src_writer = pdfrw.PdfReader(src_pdf)
    dest_writer = pdfrw.PdfReader(dest_pdf)

    src_outlines = flatten_outlines(src_reader.getOutlines ())
    dest_outlines = flatten_outlines(dest_reader.getOutlines())

    src_outline_page_nums = {outitem.title: src_reader. getDestinati
onPageNumber(outitem) for outitem in src_outlines}

    dest_outline_page_nums = {outitem.title: dest_reader.getDestinat
ionPageNumber(outitem) for outitem in dest_outlines}

    for title, src_page_num in src_outline_page_nums.items():
        if title in dest_outline_page_nums:
            dest_page_num = dest_outline_page_nums[title]
            src_page = src_writer.pages[src_page_num]
            dest_page = dest_writer.pages[dest_page_num]

            if pdfrw.PdfName.Annots not in dest_page:
                dest_page[pdfrw.PdfName.Annots] = []

            if pdfrw.PdfName.Annots in src_page:
                dest_page[pdfrw.PdfName.Annots].extend(src_page[pdfrw.
PdfName.Annots])

    pdfrw.PdfWriter().write(output_pdf, dest_writer)
```

①
②
③
④
⑤
⑥
⑦

引数は以下の通りである。

- src_pdf：アノテーションをコピーする元の PDF ファイルのパス
- dest_pdf：アノテーションをコピーする先の PDF ファイルのパス
- output_pdf：結果として出力される PDF ファイルのパス

処理は、以下の流れとなる。

① PyPDF2.PdfFileReader を使用してリーダーオブジェクトを、pdfrw.PdfReader を使用してライターオブジェクトを作成する。

② getOutlines()を使用して、それぞれの PDF ファイルからアウトライン（しおり）を取得する。その後、flatten_outlines 関数を使用してアウトラインを一次元配列に変換する。

③ getDestinationPageNumber()を使用して、コピー元とコピー先の PDF ファイル

からアウトラインのタイトルをキーとして、対応するページ番号を格納した辞書変数をそれぞれ作成する。

④ ③で作成した コピー元の辞書変数にキー情報として設定されたタイトルが、コピー先の辞書変数のキー情報に存在する場合（同一のタイトルを有する場合)、次の処理を行う

- 対応するページ番号を取得
- コピー元、コピー先において該当するページのページオブジェクトを取得

⑤ ④で取得したコピー先のページオブジェクトにアノテーションが存在しない場合、新しい空のアノテーションリストを作成する。

⑥ ④で取得したコピー元のページオブジェクトにアノテーションが存在する場合、それらをコピー先のページオブジェクトのアノテーションリストに追加する（コピー元のアノテーションを、コピー先のアノテーションに複製する)。

⑦ ⑥で取得した情報を pdfrw.PdfWriter を使用して、コピー先 PDF ファイルに書き出す。

3) 実行環境

Objects	Version
Python	3.8.10
PyPDF2	1.26.0
pdfrw	0.4
tkinter	8.6

参考文献

1) PyPDF2

PDF ファイルを操作するための python パッケージ。ページの抽出や追加、PDF の分割や結合、テキストやメタデータの読み取り等の基本的な PDF 操作を行うことができる。
https://pypdf2.readthedocs.io/en/3.0.0/

2) pdfrw

PDF ファイルを操作するための python パッケージ。PyPDF2 と同様に、ページの抽出、結合、注釈の追加、フォームフィールドの操作等が可能だが、pdfrw は PDF のコンテンツをより直接的に操作する機能を持つ。
https://github.com/pmaupin/pdfrw

2.1.3 時系列売上予測を行う機械学習モデルを Python で作成

1) 機能概要

製薬企業の営業活動では、各医療機関の売上のトレンドを把握し、減少傾向にある場合には原因を調査し対応を行うことは重要である。事例として提供しているプログラムは、医療機関ごとの日次の売上情報を元に、機械学習の手法を用いて医療機関ごとの売上を予測し、任意の期間（本例事例では 1 年に設定）の売上予測データを作成するものである。あわせて、各医療機関の売上トレンドを時系列予測する機械学習モデル作成機能を疑似的に実行するために、インプットとなる医療機関ごとの日次の売上情報を模したダミーデータ作成機能も含めている。

売上の時系列予測には prophet ライブラリを利用している。prophet は時系列データを予測するライブラリで、非線形のトレンドを年・週・日単位の周期性や祝日効果など変化点を考慮したうえでモデルの作成を行うことができる。

2） 処理概要

ダミーデータ作成機能は、下記の列から構成される施設ごと日ごとの時系列の売上を記録したダミーデータセットのデータフレームを作成する。

hco_id (病院 id)

date (年月日)

sales (売上)

```
# ダミーデータ(データフレーム)を作成する関数
def generate_dummy_data(num_hco_id, start_date, end_date):

    date_range = pd.date_range(start_date, end_date, freq='D')

    data = {'hco_id': np.random.randint(1, num_hco_id + 1, size=len(date_rang
e) * num_hco_id),
            'date': np.tile(date_range, num_hco_id),
            'sales': np.random.randint(100, 1000, size=len(date_range) * num_h
co_id)}

    df = pd.DataFrame(data)

    return df
```

引数は以下の通りである。

- num_hco_id：ダミーデータを作成したい施設数
- start_date：売上ダミーデータ作成開始日
- end_date：売上ダミーデータ作成終了日

```
# 5 施設において 2022 年 1 年間分のダミーデータを作成
dummy_data = generate_dummy_data(5, '2022-01-01', '2022-12-31')
```

上記コードを実行すると、「図 2.1.3-1 ダミーデータ」で示したようなデータが作成される。

	hco_id	date	sales
0	4	2022-01-01	875
1	5	2022-01-02	768
2	3	2022-01-03	451
3	5	2022-01-04	457
4	5	2022-01-05	616
...	...	...	...
1820	2	2022-12-27	598
1821	4	2022-12-28	853
1822	2	2022-12-29	564
1823	5	2022-12-30	790
1824	1	2022-12-31	830

[1825 rows x 3 columns]

図 2.1.3-1 ダミーデータ

医療機関ごとの日次売上情報を元に、prophet パッケージを使用し、時系列予測モデルを作成し、医療機関ごとの日次売上情報から向こう 1 年間の施設ごとの売上予測データを作成する。

```
# 施設ごとに時系列予測を行う機械学習モデルを作成する関数
def run_prophet_by_hco_id(df):
    forecast_results = []

    # Iterate over unique hco_id values
    for hco_id in df['hco_id'].unique():

        # Filter data for the current hco_id
        hco_data = df[df['hco_id'] == hco_id].reset_index(drop=True)

        # Prepare data for Prophet
        prophet_data = hco_data.rename(columns={'date': 'ds', 'sales': 'y'})

        # Initialize Prophet model
        model = Prophet()

        # Fit the model
        model.fit(prophet_data)

        # Create a future dataframe for prediction
        future = model.make_future_dataframe(periods=365)  # Adjust periods as
needed

        # Make predictions
        forecast = model.predict(future)
```

```
        # Store the forecast result for the current hco_id
        forecast['hco_id'] = hco_id
        forecast_results.append(forecast[['hco_id', 'ds', 'yhat', 'yhat_lower
', 'yhat_upper']])

    return pd.concat(forecast_results, ignore_index=True)
```

引数は以下の通りである。

・　df：病院ごとの日次売上データ（データフレーム）

```
# 機械学習モデルを作成
forecast_results_by_hco_id = run_prophet_by_hco_id(dummy_data)
```

上記コードを実行することで、引数に設定した医療機関ごとの日次売上情報から向こう 1 年間の売上予測データを作成する事ができる。

作成した売上予測に対して、実際の売上と比較することによって乖離があるかどうかを分析し、乖離がある場合は営業担当者にフィードバックすることができる。

1）　実行環境

Objects	Version
Python	3.12.0
prophet	1.1.5
pandas	2.0.3
numpy	1.25.2

参考文献

1）Prophet

https://facebook.github.io/prophet/docs/quick_start.html

2）pandas

https://pandas.pydata.org/

3）numpy

https://numpy.org/ja/

謝辞

本項の作成にあたりご助言及び資料提供いただきました、日本イーライリリーの三木康裕

様に感謝申し上げます。

2.1.4　SAS transport ファイル（.xpt）中の NonASCII 文字を検出する

1）機能概要

SAS transport ファイル中に含まれる NonASCII 文字を検出し、データセット名、変数名、行番号と共に NonASCII 文字が含まれる変数値を Excel ファイルに一覧で出力する。

2）処理概要

指定したフォルダ内に含まれる SAS transport ファイルを xport パッケージの to_dataframe() 関数を利用して pandas データフレームに変換する。データフレームに含まれる変数値を各変数及び各行ごとにループ処理で取得し、変数値ごとに NonASCII の判定を行う。

```
def find_nonascii_in_xpt(file_path):
    nonascii_records = []

    with open(file_path, 'rb') as f:
        xpt_data = xport.to_dataframe(f)

        for column in xpt_data.columns:
            for index, value in xpt_data[column].items():
                if isinstance(value, str) and is_nonascii(value):
                    nonascii_records.append((os.path.basename(file_path), co
lumn, index + 1, value))

    return nonascii_records
```

NonASCII の判定を行う際は以下の関数（is_nonascii()）を利用した。

```
def is_nonascii(text):
    return not all(ord(c) < 128 for c in text)
```

この関数では、引数で与えられた文字列を一文字ずつ切り取り、ord 関数を使って Unicode コードポイントを取得する。Non-ASCII 文字を含む場合（Unicode コードポイントが 128 未満）が一文字でも Non-ASCII 文字を含んでいる場合に True を返す。

最終的に NonASCII 文字を含むと判定された変数値について、データセット名、変数名、行番号と共に Excel ファイルに出力する。

3）実行環境

Objects	Version
Python	3.8.10
Pandas	1.3.5
xport	3.6.1
openpyxl	2.6.2
tkinter	8.6

参考文献

1) xport
SAS transport ファイル形式を読み書きするための Python のパッケージ
https://github.com/selik/xport

2.2 R 活用事例紹介

2.2.1 Microsoft Word ファイル中の表をデータフレームとして読み込む

1) 機能概要

R の docxtractr パッケージ[1)]を使って、拡張子が docx の Microsoft Word ファイル中の表を R のデータフレームとして読み込む。

2) 処理概要

拡張子が docx の Microsoft Word ファイル（以下 docx ファイル）の実体とは、XML ファイルを ZIP ファイルとして圧縮したものであり[2)]、Microsoft Word 以外のソフトウェアでもファイルの操作が可能である。R の docxtractr パッケージは、docx ファイルの XML 構造を利用し、docx ファイルから様々な情報を R に読み込むことができる[1) 3)]。

以下は PHUSE が公開しているテンプレート[4)]に従って記載した ADRG の docx ファイルから、Issue Summary の説明のテーブルを R のデータフレームとして読み込むプログラムの一例である。

```
library(tidyverse)
library(docxtractr)

doc.table <- read_docx(path = "ADRG ファイルへのパスを記載")

#### Read the tables from xDRG ####
tbl_all <- docx_extract_all_tbls(docx = doc.table)

tbl_cnt <- docx_tbl_count(doc.table)
```

```
tblnm_out <- c(1:tbl_cnt)

for (i in tblnm_out) {
  tblnm_out[i] <- tbl_all[[i]] %>%
    gather(category, value) %>%
    filter(category=="Diagnostic.Message") %>%
    summarise(n())
}

tblnm_out2 <- unlist(tblnm_out)
adrg <- bind_rows(tbl_all[which(tblnm_out2>0)])
```

テーブルの内容を R のデータフレームにできれば、R 内で表中のデータの操作が可能となる。また、Microsoft Excel の形式に出力して Excel 内でデータ操作を行える。

3) 実行環境

Objects	Version
R	4.3.0
docxtractr	0.6.5

参考文献

1) docxtractr: Extract Data Tables and Comments from 'Microsoft' 'Word' Documents
https://cran.r-project.org/web/packages/docxtractr/index.html

2) Open XML Formats とファイル名拡張子
https://support.microsoft.com/ja-jp/office/open-xml-formats-%E3%81%A8%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E5%90%8D%E6%8B%A1%E5%BC%B5%E5%AD%90-5200d93c-3449-4380-8e11-31ef14555b18

3) R が生産性を高める～データ分析ワークフロー効率化の実践　技術評論社
書籍

4) Analysis Data Reviewer's Guide (ADRG) Package
https://advance.phuse.global/display/WEL/Analysis+Data+Reviewer%27s+Guide+%28ADRG%29+Package

2.2.2 haven パッケージによる各種の統計ソフトウェア用ファイルの入出力

1) 機能概要

R の haven パッケージ [1)]を使って、SAS、SPSS、Stata 用のファイルを入出力する。

2） 処理概要

R の haven パッケージを使うことで、SAS、SPSS、Stata 用のデータ形式のファイルをデータフレームとして読み込むことができる。逆に、R のデータフレームを、SAS、SPSS、Stata 用のデータ形式のファイルとして作成することもできる。haven のバージョン 2.5.3 が対応しているファイルの形式は「表 2.2.2-1 haven が対応しているファイル形式」の通りである。

統計ソフトウェア	ファイル形式（拡張子）	読み込み	作成
SAS	SAS transport ファイル（.xpt）	可能	可能
SAS	SAS データセット（.sas7bdat）	可能	不可能
SPSS	IBM SPSS Statistics データ・ファイル（.sav, .zsav）	可能	可能
Stata	Stata DTA ファイル（.dta）	可能	可能

表 2.2.2-1 haven が対応しているファイル形式

SAS transport ファイルはバージョン 5 とバージョン 8 の両方の入出力が可能である。また、haven は Tidyverse パッケージの一部にもなっているため、Tidyverse をインストールすれば利用可能になる [2)]。以下は、R Consortium による R を使った FDA への Pilot 申請で提出された ADaM の SAS transport ファイルをデータフレームとして読み込み、データフレームのサブセットを SAS transport ファイルに出力するサンプルプログラムである。

```
library(haven)
filename <- "https://github.com/RConsortium/submissions-pilot3-adam-to-fda/raw/main/m5/datasets/rconsortiumpilot3/analysis/adam/datasets/adadas.xpt"

adadas <- read_xpt(filename)
adactot <- subset(adadas, PARAMCD=="ACTOT")
write_xpt(adactot, "adactot.xpt", version=5)
```

3） 実行環境

Objects	Version
R	4.3.0
haven	2.5.3

参考文献

1) haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files
https://cran.r-project.org/web/packages/haven/index.html

2) tidyverse: Easily Install and Load the 'Tidyverse'
https://cran.r-project.org/web/packages/tidyverse/index.html

2.2.3　datasetjson パッケージによる Dataset-JSON 形式のデータセットの入出力

1）機能概要

R の datasetjson パッケージ [1)]を使って、CDISC の Dataset-JSON 形式のデータセットを入出力する。

2）処理概要

SAS transport ファイルのバージョン 5 は SAS Institute Inc.が定義したデータのファイル形式であり、仕様が一般に公開されているため、FDA や PMDA に申請電子データを提出する際の SDTM と ADaM のファイル形式として採用されている。しかし、1980 年代に作られた仕様であるため、変数名が 8 文字以内、変数ラベルが 40 文字以内、データの長さが 200 文字以内という制限があり、これらの条件に合わせて申請時に提出する SDTM と ADaM を作る必要があった。このため、CDISC では SDTM と ADaM を、より仕様の自由度が高い XML 形式で作成するための標準、「Dataset-XML」を公開した。ところが、SDTM や ADaM を XML 形式で作成すると、データにタグとして説明を記載する必要があるためか、ファイルサイズが増大するという問題があった [2)]。

そこで、CDISC が新たに公開した標準が「Dataset-JSON」である [3)]。JSON とは、XML と同様、アプリケーション間のデータ交換に使用されるデータ表現であるが、構文がよりコンパクトという特徴があり、Dataset-JSON は SDTM と ADaM を JSON 形式で作成するための仕様の標準である。R には、もともと JSON 形式のファイルの入出力を行う jsonlite パッケージ [4)]もあったが、Dataset-JSON に特化した datasetjson パッケージが新たに公開された。以下は、github 上に公開された Dataset-JSON 形式の ADaM データセットを読み込み、R のデータフレームを Dataset-JSON 形式のファイルに出力するサンプルプログラムである（抜粋）。Dataset-JSON 形式のファイルを出力する際は変数名等のメタデータを dataset_json 関数で指定する必要があるが、このサンプルプログラムでは jsonlite パッケージで Dataset-JSON 形式のファイルから読み込んだメタデータを使用している。

```
library(datasetjson)
library(jsonlite)
library(haven)
```

```
#### Read the contents from Dataset-JSON file ####
jsonfilename <- "https://raw.githubusercontent.com/cdisc-org/DataExchange
-DatasetJson/master/examples/adam/adadas.json"
json_data <- read_dataset_json(jsonfilename)

#### Read the contents from SAS transport file (.xpt) file ####
xptfilename <- "https://raw.githubusercontent.com/cdisc-org/DataExchange-
DatasetJson/master/examples/adam/adadas.xpt"
xpt_data <- read_xpt(xptfilename)

#### Write the contents from SAS transport file (.xpt) file into Dataset-J
SON file ####
json_metadata <- fromJSON(jsonfilename)
xpt_json <- dataset_json(xpt_data,
                         "IG.ADADAS",
                         "ADADAS",
                         "ADAS-Cog Analysis",
                         json_metadata$clinicalData$itemGroupData$IG.ADAD
AS$items)

write_dataset_json(xpt_json,"xpt_json.json")
```

3) 実行環境

Objects	Version
R	4.1.2
datasetjson	0.2.0
jsonlite	1.8.4

参考文献

1) datasetjson: Read and Write CDISC Dataset JSON Files
https://cran.r-project.org/web/packages/datasetjson/index.html

2) History of Submission Data Formats
https://www.atorusresearch.com/datasetjson-0-0-1-release/

3) Dataset-JSON
https://www.cdisc.org/dataset-json

4) jsonlite: A Simple and Robust JSON Parser and Generator for R
https://cran.r-project.org/web/packages/jsonlite/index.html

2.2.4　R Shiny による複数のがん第 1 相試験デザインの動作特性の比較検討

1）　機能概要

R Shiny[1)]および R の escalation パッケージ[2)]を使って、GUI（Graphical User Interface）環境下で複数のがん第 1 相試験デザインの動作特性を比較検討する。
R Shiny で Web アプリを開発する際、パラメータ設定（ユーザインタフェース）と、データ処理に分けてプログラムを作成する必要がある。本事例では、データ処理の BOIN によるシミュレーション実装部分について紹介する。

2）　処理概要

R の escalation パッケージでは、がん第 1 相試験における種々の用量漸増デザインを対象とした動作特性シミュレーションの実装が可能である。
以下にシミュレーションの実装に必要となる主な入力パラメータおよびシミュレーションの結果、出力されるパラメータの一覧を示す。「表 2.2.4-1 入力パラメータ」では、例として用量漸増デザインは、Model-based デザインから CRM（Continual Reassessment Method）、Model-assisted デザインから BOIN（Bayesian optimal interval design）の 2 手法を提示している。escalation パッケージに搭載されているその他の手法の一覧および各手法の詳細については、パッケージの Documentation[2)]を参照されたい。

主な入力パラメータ		主な出力パラメータ
全般的な設定	・　各用量における真の毒性発現確率 ・　標的毒性確率 ・　用いる用量漸増デザイン ・　サンプルサイズ ・　シミュレーション回数	・　用量ごとの MTD（Maximum Tolerated Dose）選択割合 ・　用量ごとの投与割合 ・　DLT（Dose Limiting Toxicity）発現例数
手法ごとの設定例	CRM ・　スケルトンの設定 ・　作業モデルの設定 BOIN ・　標的毒性確率の下側限界値 ・　標的毒性確率の上側限界値	

表 2.2.4-1 入力パラメータ

また、コード例として、以下に BOIN によるシミュレーション実装部分のサンプルコードを提示する。

```
# ライブラリのロード
library(tidyverse)
library(escalation)

# 標的毒性確率の設定
target_tox <- 0.25
# 標的毒性確率の下側限界値
target_tox_ll = 0.6 * target_tox
# 標的毒性確率の上側限界値
target_tox_ul = 1.4 * target_tox
# 用量水準数の設定
num_doses <- 6
# 各用量における真の毒性発現確率の設定
true_prob_tox <- c(0.25, 0.35, 0.5, 0.6, 0.7, 0.8)
# 最大サンプルサイズの設定
max_n <- 36
# シミュレーション回数の設定
num_sims <- 50

# BOIN モデルの作成
model <- get_boin(num_doses = num_doses, target = target_tox, p.saf = targ
et_tox_ll, p.tox = target_tox_ul) %>%
  stop_at_n(n = max_n)

#シミュレーションの実装
sims <- model %>%
  simulate_trials(num_sims = num_sims, true_prob_tox = true_prob_tox)
```

上記の動作特性シミュレーションを、R Shiny を用いてアプリケーション化し、複数の手法を同時に比較検討できるツールを構築した。以下に、実際のツールの入力パラメータの設定画面およびシミュレーション結果の出力画面の一部を示す。

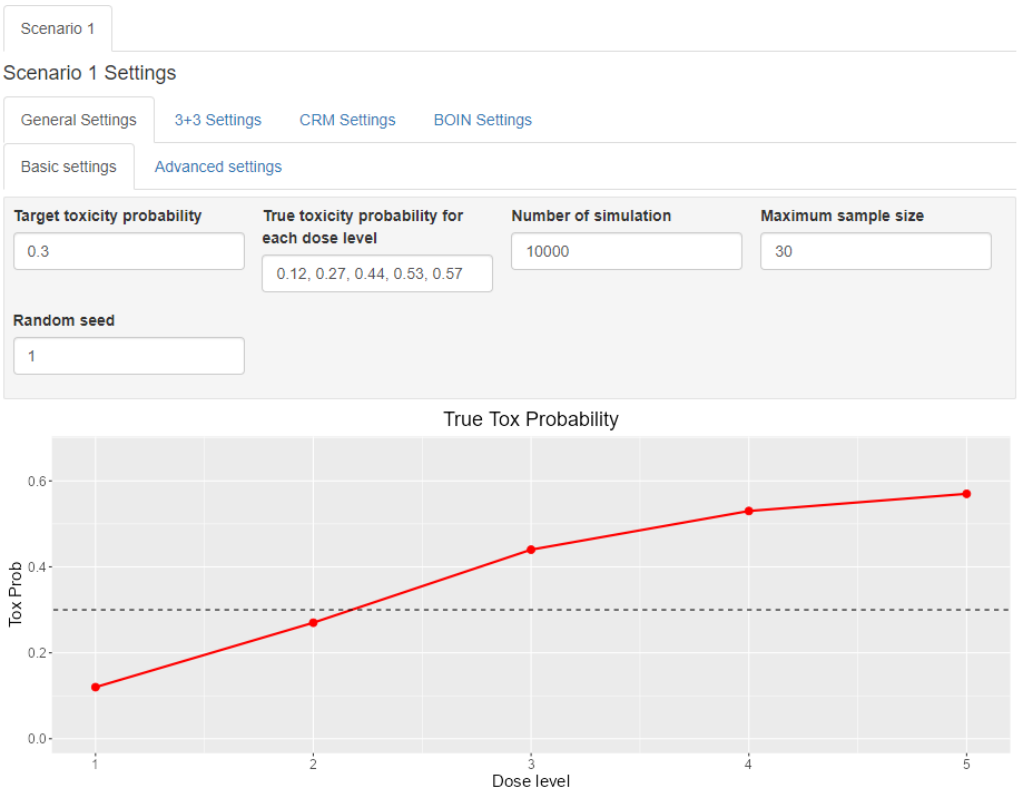


図 2.2.4-1 入力パラメータの設定画面

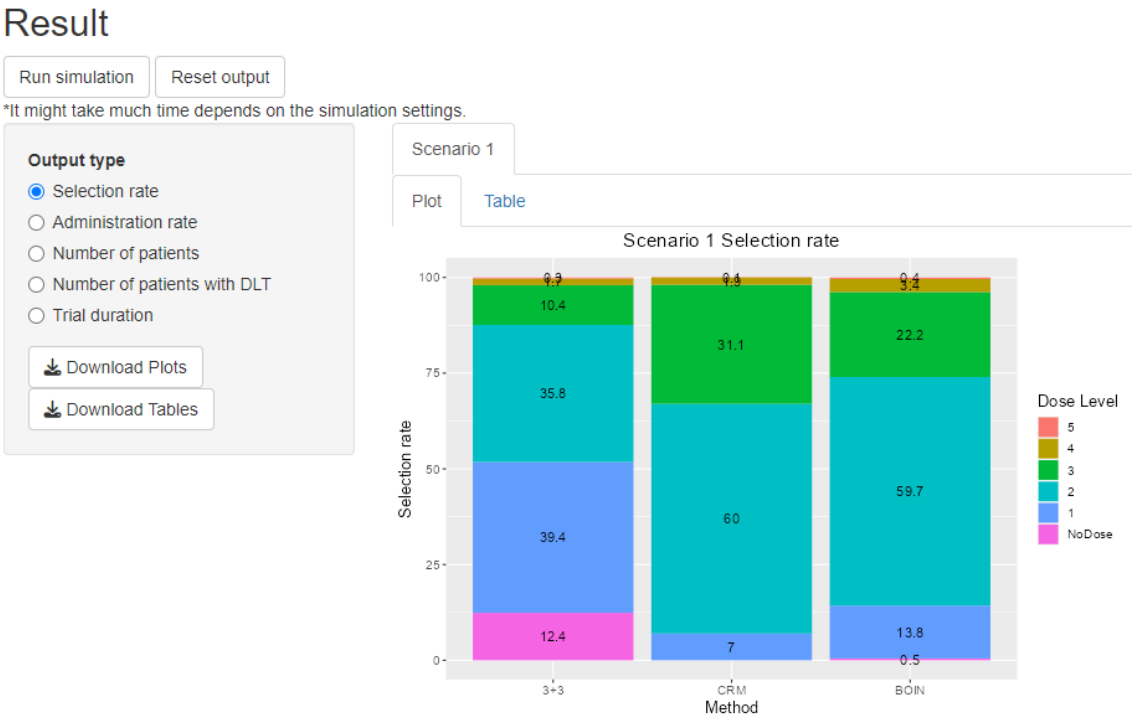


図 2.2.4-2 出力画面のキャプチャ

がん第 1 相試験の用量漸増デザインは近年多くの手法が提案されており [2)]、薬剤の特性や開発戦略に応じて、適したデザインを選択することが重要である。本ツールを用いることで、それら複数の手法に対して様々なシナリオ下での動作特性の比較評価が可能となり、がん第 1 相試験の検討の際に役立てることができる。また、GUI での操作および結果の提示は直観的にも理解しやすく、統計担当者以外のメンバーとの議論の際にも有用である。

3) 実行環境

Objects	Version
R	4.3.0
escalation	0.1.5

参考文献

1) Shiny

R 言語で web アプリケーションを簡単に作成出来るパッケージである。Shiny アプリを動かすには、基本的な構造として、パラメータ操作のための「user interface」とデータ処理のための「server」の 2 つのファイルを作る必要がある。

https://shiny.posit.co/

2) escalation: Modular Approach to Dose Finding Clinical Trials

https://brockk.github.io/escalation/

3) 近年のがん第 I 相試験デザインとその選択

https://www.jpma.or.jp/information/evaluation/results/allotment/DS_202306_oncoP1DE.html

2.2.5 RTF ファイルをプレーンテキストとして読み込む

1) 機能概要

R のパッケージ「striprtf」を使って、Rich Text Format ファイル（以下 RTF ファイル）をプレーンテキストとして読み込む。

2) 処理概要

RTF ファイルは、プレーンテキストに文字のフォント・大きさ・色や太字・斜体などの装飾指定、表などの簡易レイアウトのための制御用文字列を付加した形式となって

いるが、R の striprtf パッケージは、RTF ファイルをプレーンテキストとして読み込むことができる。
以下は、指定したディレクトリ下に存在する RTF ファイルを特定し、指定した RTF ファイルの内容をまるごと読み込むプログラム例である。

```
library(striprtf)

#Set the path for RTF file
path <- c("¥¥server¥")

#Specify RTF file name to read
outputname <- c("table1","table2","table3")

#ファイル名
filesrtf <- paste(outputname,".rtf",sep="")
filesrtffull <- paste(path,filesrtf,sep="")

#Loop over RTF files
for(i in seq(length(filesrtffull))){
  aFile <- filesrtffull[i]
  if (! file.exists(aFile)){
    cat(filesrtf[i],"not exist¥n")
    next
  }
  cat(filesrtf[i],"¥n")
  t2 <- read_rtf(filesrtffull[i])

  #Display the read text from RTF file
  print(t2)
}
```

RTF ファイルの内容をプレーンテキストとして読み込むことができれば、読み込んだデータからさらに必要な情報を取り出すなどの操作も行うことが可能になる。
臨床試験データの解析帳票が RTF ファイルであれば、帳票中の解析結果を R のデータフレームとして読み込むことで、必要な解析結果を抽出し、他の解析帳票のためのデータとするなど二次利用に用いることも可能となる。

3) 実行環境

Objects	Version
R	4.3.0
striprtf	0.6.0

参考文献

1) striprtf: Extract Text from RTF File

https://cran.r-project.org/web/packages/striprtf/striprtf.pdf

2.2.6　回帰分析を用いてマーケティング活動の効果を評価する統計モデルを R で作成

1）　機能概要

マーケティング活動を効果的に行うためには、各プロモーションチャネル（MR 活動、メール、講演会）の効果を定量的に評価し、より費用対効果の高いプロモーションチャネルに投資配分を最適化していくことが重要である。

このようなマーケティング活動を効果測定するための統計モデルとして MMM (Marketing Mix Modelling)[1)]が知られている。本事例では売上を目的変数、プロモーション活動の量を説明変数とした MMM を実施する際のダミーデータ作成と回帰分析の R コードのサンプルを提供する。なお、MMM では様々なモデルがあり、実際には、様々なモデルを試したり、説明変数の取捨選択をしたり、変数間の関係性などを深く検討する必要がある点にご留意ください。

2）　処理概要

下記の項目から構成される毎月の病院ごとの売上とマーケティング活動の回数を記録したダミーデータセットを 1 年分作成する。

hco_id (病院 id)
year_month (年月)
rep_activity (MR 訪問の回数)
email (メールの開封数)
web_P2P (ウェブ講演会の視聴数)
local_P2P (地区講演会の参加回数)
edetail (外部医療メディアコンテンツの視聴回数)

```
# 結果の再現性のため seed を設定
set.seed(123)

# ダミーデータセットの病院の数
num_hco_id <- 1000
# ダミーデータセットの期間(月数)
num_months <- 24

# 期間(月数)と病院の数の総組み合わせを作成
hco_id <- rep(1:num_hco_id, each = num_months)
year_month <- rep(seq(as.Date("2022-01-01"), length.out = num_months, by
= "months"), times = num_hco_id)

# ダミーデータ(データフレーム)を作成
data <- data.frame(
```

```
  hco_id = hco_id,
  year_month = year_month,
  rep_activity = rpois(num_hco_id * num_months, lambda = 30),
  email = rpois(num_hco_id * num_months, lambda = 20),
  web_P2P = rpois(num_hco_id * num_months, lambda = 15),
  local_P2P = rpois(num_hco_id * num_months, lambda = 10),
  edetail = rpois(num_hco_id * num_months, lambda = 25)
)
```

上記プログラムでは、病院 1000 施設に対して 2022 年 1 月 1 日から 1 年間分のマーケティング活動回数のダミーデータを作成している。

また、各チャネルの 1 回あたりの月次の売上（sales）への効果を下記のように想定してダミーデータを作成し、モデルを評価する。
rep_activity（MR 訪問の回数）：8000 円/回
email（メールの開封数）：1000 円/回
web_P2P（ウェブ講演会の視聴数）：5000 円/回
local_P2P（地区講演会の参加回数）：10000 円/回
edetail（外部医療メディアコンテンツの視聴回数）：1500 円/回

```
# 売上ダミーデータ列の作成
data$sales <- 8000 * data$rep_activity +
              1000 * data$email +
              5000 * data$web_P2P +
              10000 * data$local_P2P +
              1500 * data$edetail +
              # ダミーデータ作成のため売上のばらつき分を追加
              rnorm(num_hco_id * num_months, mean = 0, sd = 10000)
```

作成した病院 1000 施設分の 1 年間分のマーケティング活動回数と売上データをもとに、MMM の回帰モデルを作成し、summary 関数で結果を確認する。

```
# MMM の回帰分析モデルを作成
mmm_model <- lm(sales ~ rep_activity + email + web_P2P + local_P2P + edeta
il, data = data)

# model result summary
summary(mmm_model)
```

summary 関数を実行すると、「図 2.2.6-1 MMM によりマーケティング活動効果測定結果」のような結果が出力される。出力から次のように解釈できる。

```
Call:
lm(formula = sales ~ rep_activity + email + web_P2P + local_P2P +
    edetail, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-44141  -6723     35   6828  38594

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     48.06     646.51   0.074    0.941
rep_activity  8002.79      11.79 678.530   <2e-16 ***
email          988.78      14.31  69.085   <2e-16 ***
web_P2P       5008.87      16.60 301.779   <2e-16 ***
local_P2P    10013.80      20.42 490.390   <2e-16 ***
edetail       1491.23      12.81 116.395   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9987 on 23994 degrees of freedom
Multiple R-squared:  0.9712,    Adjusted R-squared:  0.9712
F-statistic: 1.616e+05 on 5 and 23994 DF,  p-value: < 2.2e-16
```

図 2.2.6-1 MMM によりマーケティング活動効果測定結果

- 各プロモーション活動の売上への相関はすべて統計的に有意であり、プロモーション活動による売上へのインパクトを期待できる。
- 上記のダミーデータ作成で設定した通り、1 回各活動を実施したときの月次の売上への効果は rep_activity : 8000 円/回、email : 1000 円/回、web_P2P : 5000 円/回、local_P2P : 10000 円/回、edetail : 1500 円/回と推定されている。これは、例えば rep_activity を 1 回行うと、その施設のその月の売上が追加で 8000 円上昇するということを意味している。

この結果をもとに費用対効果を評価するためには、各活動を実施した際の 1 回あたりの費用も別のデータソースを用いて計算しておき、ROI (Return on Investment)を計算することで評価できる。例えば、rep activity1 回あたりの費用が 6000 円/回であれば ROI は 8000/6000 = 1.3 と評価でき、投資を上回る効果をあげられていると解釈できる。

3) 実行環境

Objects	Version
R	4.2.2
stats	4.2.2

参考文献

1） Marketing Mix Modeling (MMM)- Concepts and Model Interpretation
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3877291

謝辞

本項の作成にあたりご助言及び資料提供いただきました、日本イーライリリーの三木康裕様に感謝申し上げます。

2.2.7　tidytlg パッケージを用いたテーブル、リスト、グラフの生成

1）機能概要

解析帳票のテーブル、リスト、グラフ（TLG）を作成するためには、大きく分けて下の 4 つのステップがある。

1. 環境準備
   入出力の R 環境をセットアップ
2. データの処理
   分析データをフィルタリング、データ操作(文字変数をファクタに変換する等)、列変数を定義
3. 結果の生成
   集計表や、グラフを作成
4. 結果出力
   3.で作成した解析結果を rtf や html の形式で出力する。

本事例ではこれらのフレームワークを、tidytlg パッケージを用いて実現する方法について紹介する。

2）処理概要

tidytlg パッケージによるテーブル作成について、「図 2.2.7-1 出力帳票イメージ」を例に手順を説明する。

Table01: Demographic and Baseline Characteristics; Intent-to-treat Analysis Set

	Xanomeline				
	Placebo	Low Dose	High Dose	Combined	Total
Analysis set: ITT	86	84	84	168	254
Age, years					
N	86	84	84	168	254
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)
Median	76.0	77.5	76.0	77.0	77.0
Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)
Gender					
N	86	84	84	168	254
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)	78 (46.4%)	111 (43.7%)
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)	90 (53.6%)	143 (56.3%)
Unknown	0	0	0	0	0

Key: IQ = interquartile

Note: N reflects non-missing values

[table01.html][] 31AUG2023, 15:58

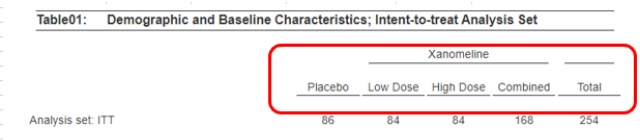
図 2.2.7-1 出力帳票イメージ

1. 環境準備

   入出力ファイルのパスを設定。ADaM と、帳票のタイトルファイル(titles.xlsx)や列構造を定義するファイル(column_metadata.xlsx)等は入力フォルダに配置し、出力フォルダは出力ファイルの保存に使用する。

2. データの処理（ファンクション：tlgsetup）

   集計時の列データとなる各群や合計群の情報を Excel ファイルで設定する。



   プログラムにて tbltype を指定することにより、Excel ファイルから列情報を取得する。

```
adsl01 <- adsl %>%
  filter(ITTFL == "Y") %>%
  mutate(SEX = factor(SEX, levels = c("M", "F", "U"),
                           labels = c("Male", "Female", "Unknown"))) %>%
  tlgsetup(var = "TRT01PN",
           column_metadata_file = system.file("extdata/column_metadata.xlsx", package = "tidytlg"),
tbltype = "type3")
```

[column_metadata.xlsx]

テーブルレイアウトの列構造を定義するファイル。

tbltype: テーブルの列レイアウトの識別子

coldef: 投与群の変数，例では TRT01PN の値

decode: テーブルの列ヘッダーの表示名

span1-3: テーブルの階層型列ヘッダーの表示名

	A	B	C	D	E	F
1	**tbltype**	**coldef**	**decode**	**span1**	**span2**	**span3**
2	type1	0	Placebo			
3	type1	54	Low Dose	Xanomeline		
4	type1	81	High Dose	Xanomeline		
5	type2	0	Placebo			
6	type2	54	Low Dose	Xanomeline		
7	type2	81	High Dose	Xanomeline		
8	type2	54+81	Combined	Xanomeline		
9	type3	0	Placebo			
10	type3	54	Low Dose	Xanomeline		
11	type3	81	High Dose	Xanomeline		
12	type3	54+81	Combined	Xanomeline		
13	type3	0+54+81	Total			

[出力- adsl01 データフレーム]

カラム情報となる tbltype, colnbr 変数が追加される。

tbltype	colnbr	TRT01PN	USUBJID
type3	col1	0	A
type3	col2	54	B
type3	col3	81	C
type3	col4	54	D
type3	col1	0	E

3. 結果の生成（ファンクション：univer/freq）

tlgsetup で定義された群単位での要約統計量、頻度集計を算出する。統計量やフォーマットを指定することにより集計結果を取得できる。

Table01: Demographic and Baseline Characteristics; Intent-to-treat Analysis Set

		Xanomeline			
	Placebo	Low Dose	High Dose	Combined	Total
Analysis set: ITT	86	84	84	168	254
Age, years					
N	86	84	84	168	254
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)
Median	76.0	77.5	76.0	77.0	77.0
Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)
Gender					
N	86	84	84	168	254
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)	78 (46.4%)	111 (43.7%)
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)	90 (53.6%)	143 (56.3%)
Unknown	0	0	0	0	0

Key: IQ = interquartile

Note: N reflects non-missing values

```
## Analysis set row
t1 <- adsl %>%
```

```
  freq(colvar = "colnbr",
       rowvar = "ITTFL",
       statlist = statlist("n"),
       subset = ITTFL == "Y",
       rowtext = "Analysis set: ITT")

## Univariate summary for AGE
t2 <- adsl %>%
  univar(colvar = "colnbr",
         rowvar = "AGE",
         statlist = statlist(c("N", "MEANSD", "MEDIAN", "RANGE", "IQRANGE
")),
         decimal = 0,
         row_header = "Age, years")

## Count (percentages) for SEX
t3 <- adsl %>%
  freq(colvar = "colnbr",
       rowvar = "SEX",
       statlist = statlist(c("N","n (x.x%)")),
                           row_header = "Gender")

# Format Results ----------------------------------------------------------
tbl <- bind_table(t1, t2, t3,
                  column_metadata_file = system.file("extdata/column_meta
data.xlsx", package = "tidytlg"),
                  tbltype = "type3")
```

[出力- tbl データフレーム]

label	col1	col2	col3	col4	col5	row_type	anbr	indentme	roworder	newrows	newpage
Analysis set: ITT	86	84	84	168	254	HEADER	1	0	1	0	0
Age, years						HEADER	2	0	1	1	0
N	86	84	84	168	254	N	2	1	2	0	0
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)	VALUE	2	2	3	0	0
Median	76.0	77.5	76.0	77.0	77.0	VALUE	2	2	4	0	0
Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)	VALUE	2	2	5	0	0
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)	VALUE	2	2	6	0	0
Gender						HEADER	3	0	1	1	0
N	86	84	84	168	254	N	3	1	2	0	0
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)	78 (46.4%)	111 (43.7%)	VALUE	3	2	3	0	0
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)	90 (53.6%)	143 (56.3%)	VALUE	3	2	4	0	0
Unknown	0	0	0	0	0	VALUE	3	2	5	0	0

4. 結果の出力（ファンクション：gentlg）

集計結果を指定したディレクトリに RTF ファイルとして出力する。タイトルやフットノートの情報は Excel ファイルから取得して表示する。

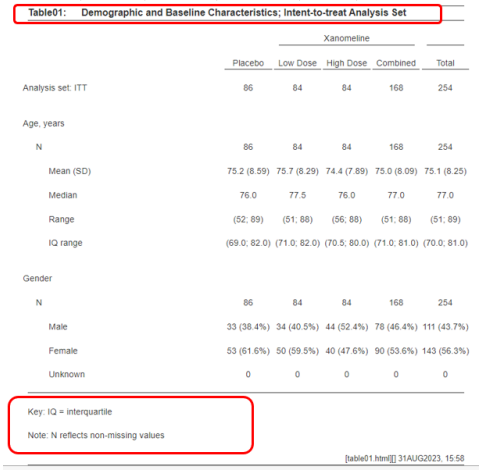


Table01: Demographic and Baseline Characteristics; Intent-to-treat Analysis Set

	Xanomeline				
	Placebo	Low Dose	High Dose	Combined	Total
Analysis set: ITT	86	84	84	168	254
Age, years					
N	86	84	84	168	254
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)
Median	76.0	77.5	76.0	77.0	77.0
Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)
Gender					
N	86	84	84	168	254
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)	78 (46.4%)	111 (43.7%)
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)	90 (53.6%)	143 (56.3%)
Unknown	0	0	0	0	0

Key: IQ = interquartile

Note: N reflects non-missing values

[table01.html][] 31AUG2023, 15:58

```
gentlg(huxme = tbl,
       opath = file.path(working_dir),
       file = "Table01",
       orientation = "landscape",
       title_file = system.file("extdata/titles.xls", package = "tidytlg
"))
```

[titles.xls]

タイトル、フットノートを Excel ファイルに設定する。

TABLE ID	IDENTIFIER	TEXT
Table01	TITLE	Demographic and Baseline Characteristics; Intent-to-treat Analysis Set
Table01	FOOTNOTE1	Key: IQ = interquartile
Table01	FOOTNOTE2	Note: N reflects non-missing values

[出力 - RTF ファイル]

RTF ファイルに集計結果が出力される。

3) 実行環境

Objects	Version
R	4.2.1
tidytlg	0.1.4

参考文献

1) pharmaverse - tidytlg

https://pharmaverse.github.io/tidytlg/main/index.html

2) CRAN - tidytlg
https://cloud.r-project.org/web/packages/tidytlg/index.html

2.2.8　R Shiny を用いた ADaM データを表示するツール

1)　機能概要

R Shiny[1)]を用いて ADaM データセットから直接データを表示する。帳票作成を行う前にデータを表示できるため、承認申請時や照会事項対応時に社内で議論の際に活用できる。更に、予備的な解析やロジックの確認も帳票作成前に行えるため、帳票作成後の再解析になるリスクを減すことも期待できる。Shiny は Web アプリとして公開できることから、プログラミングの知識が無くても GUI 上で、直感的に ADaM データの操作が行えるため、臨床担当者やメディカルライターでも比較的容易に利用できることが期待される。

通常、Shiny でアプリケーションを作成する場合、ユーザインタフェース部分（ui.R）と、データ処理部分（server.R）の 2 つのプログラムに分けて開発するが、本事例では、これら 2 つをまとめたプログラム（app.R）の事例を紹介する。

2)　処理概要

以下のプログラムはデータ処理部分のコードになる。DT[2)]パッケージの renderDataTable 関数で、admiral[3)]パッケージの ADSL サンプルデータを読み込み、アプリケーション上で「図 2.2.8-1 ADSL 出力事例」のようなインタラクティブに表示アクセスできるようにしている。

```
#Load packages
library(shiny)
library(DT)
library(admiral)

#server
server <- shinyServer(function(input, output) {
   output$table = renderDataTable(admiral::admiral_adsl, filter = "top"
      })
```

以下のプログラムはユーザインタフェース部分のコードになる。

DT パッケージの dataTableOutput 関数でデータ処理部分にて設定したデータ（table 変数）を出力している。

```
#user interface
ui <- shinyUI(fluidPage(
```

```
  titlePanel("ADSL"),
  dataTableOutput("table")
))
```

最後にデータ処理とユーザインタフェース処理で作成した変数を、Shiny パッケージの shinyApp 関数に渡してアプリケーションを実行している。

```
#run the app
shinyApp(ui, server)
```

上記のように非常に短いスクリプトで、検索、ソーティング、フィルタリングが行えるデータテーブルのアプリケーションを表示させることができる。

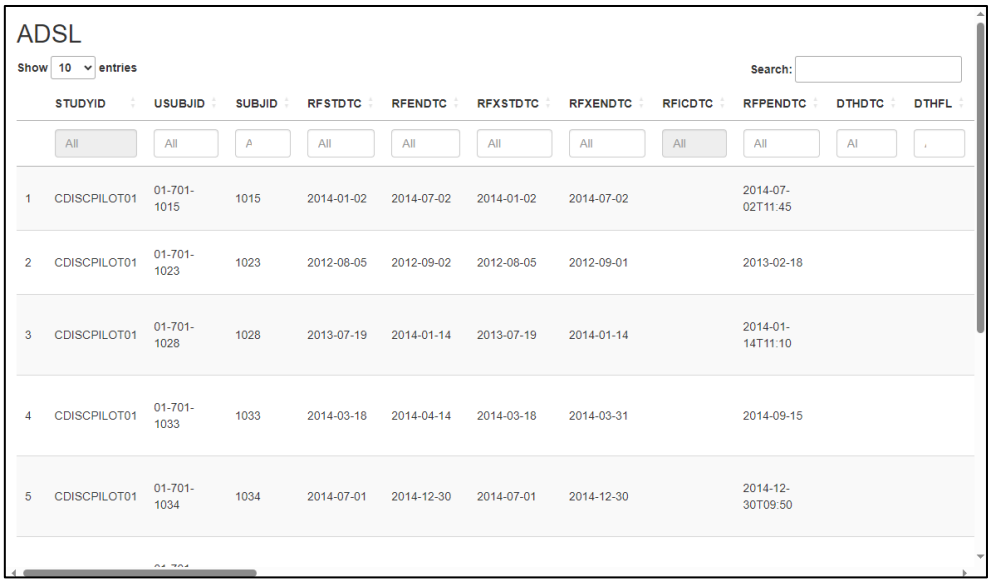


図 2.2.8-1 ADSL 出力事例

3) 実行環境

Objects	Version
R	4.1.0
shiny	1.7.4
DT	0.26
admiral	0.7.0

参考文献

1) Shiny

R 言語で web アプリケーションを簡単に作成出来るパッケージである。Shiny アプリを動かすには、基本的な構造として、パラメータ操作のための「user interface」とデータ処理

のための「server」の 2 つのファイルを作る必要がある。
https://shiny.posit.co/

2) DT
インタラクティブなデータテーブルを HTML で作成する事ができるパッケージ
https://cran.r-project.org/web/packages/DT/index.html

3) admiral
ADaM を作成するためのパッケージ。疾患領域に応じた ADaM を作成するための関数も公開している。
https://cran.r-project.org/web/packages/admiral/index.html

2.2.9　R Shiny により単群がん第 2 相試験の必要症例数を検討するツール

1)　機能概要

R Shiny および R の EnvStats パッケージ[1)]を使って、GUI（Graphical User Interface）環境下で ORR（Objective Response Rate）を主要エンドポイントとした単群がん第 2 相試験の必要症例数を検討する。

2)　処理概要

R の EnvStats パッケージの propTestPower 関数では、単群または 2 群の 2 値データに対する検定の検出力の算出が可能である。本事例では、ORR を主要エンドポイントとした単群がん第 2 相試験を想定し、propTestPower 関数を用いフィッシャーの正確検定ベースでの必要症例数を算出するツールを R Shiny を用いて構築した。以下に本ツールの入力パラメータおよび出力される数値や図の一覧を示す。

入力パラメータ	出力される数値や図
・　期待奏効率 ・　閾値奏効率 ・　α エラー ・　検出力 ・　検定の種類（片側 or 両側）	数値 ・　必要症例数 ・　試験成功基準を満たす最小奏効例数 ・　実際の検出力 ・　実際の α エラー 図 ・　期待奏効率下および閾値奏効率下での奏効例数の累積分布関数および確率質量関数

	・ 各奏効例数での信頼区間 ・ 各症例数での検出力および試験成功基準を満たす最小奏効例数

以下に実際のツールの入力パラメータの設定部分（図 2.2.9-1 入力パラメータの設定部分のキャプチャ）および症例数算出の結果の出力部分（図 2.2.9-2 出力結果の表示部分のキャプチャ）の一部を示す。

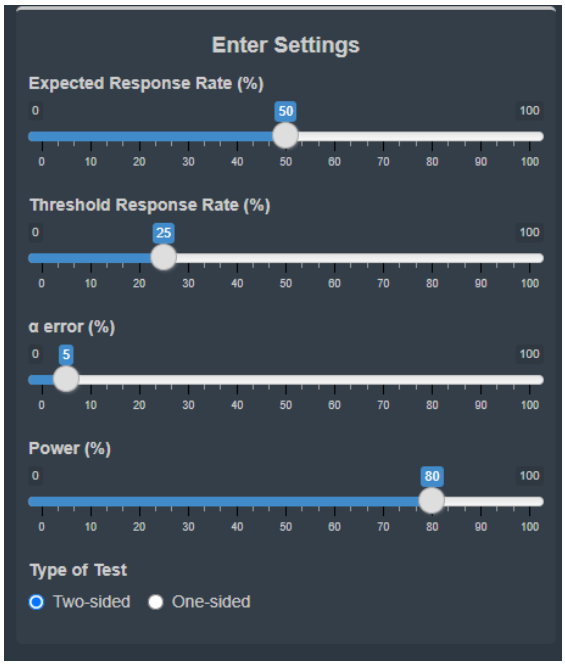


図 2.2.9-1 入力パラメータの設定部分のキャプチャ

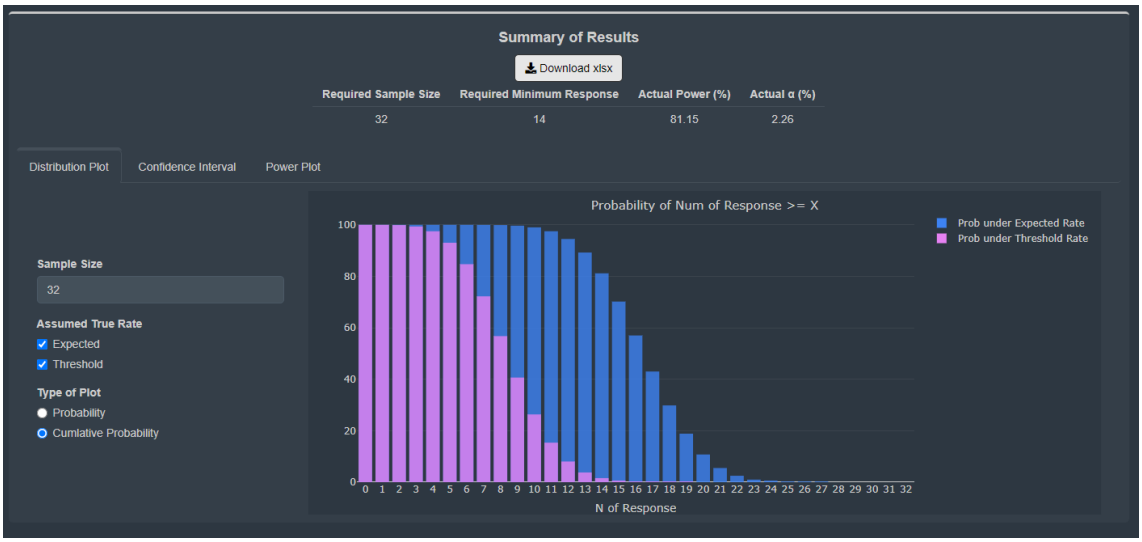


図 2.2.9-2 出力結果の表示部分のキャプチャ

単群がん第 2 相試験の実施検討にあたり、本ツールを用いることで、統計担当者以外のメンバーが探索的に必要症例数の検討を行うことが可能になると想定している。

3) 実行環境

Objects	Version
R	4.3.0
escalation	2.8.0

参考文献

1) EnvStats: Package for Environmental Statistics, Including US EPA Guidance
https://cran.r-project.org/web/packages/EnvStats/index.html

2.2.10 R Shiny によるがん第 I 相試験におけるモデル支援型デザインの動作特性を評価するツール

1) 機能概要

近年のがん第 I 相試験では、そのデザインとしてモデル支援型デザインが注目されており、特に Bayesian Optimal Interval（BOIN）デザイン [1)]が FDA の Drug Development Tools: Fit-for-Purpose Initiative[2)]により指定されている。また、がん第 I 相試験において古典的な 3+3 デザイン以外の用量漸増デザインを用いる場合、30 日調査照会事項チェックリスト（抗悪性腫瘍剤分野）[3)]に基づき、シミュレーションにより当該デザインの動作特性を示し、その適切性を PMDA に提示する必要がある。これらのシミュレーションにあたって、既存のツールではカバーされない試験特有の変更を加えることも少なくない。この場合、新たにシミュレーション用のプログラムの開発が必要になるため、効率的に動作特性を確認する体制の構築が重要となる。本ツールでは Single Patient Acceleration4)を許容した modified 3+3 デザイン及び modified BOIN について効率的な動作特性の評価及びその比較を可能とした。出力される図表のダウンロード機能を有しているため、文書作成等への活用が容易である。さらに R スクリプトのダウンロード機能を有しているため、出力結果の再現性を担保でき、本ツールでは対応できないような検討中の試験デザインに応じたコードの改変を容易に行うことができる。

2) 処理概要

真の DLT 発現確率のシナリオをまとめた CSV ファイル並びにデザインを特徴づける複数のパラメータを入力として、modified 3+3 デザイン及び modified BOIN デザインを適用した際の各用量を MTD と選択する割合や組み入れられる被験者数を出力す

る。以下に本ツールの入力パラメータ及び出力される数値や図の一覧を示す。

入力パラメータ	出力される数値や図
・ 真の DLT 発現確率のシナリオ ・ シミュレーション回数 ・ 目標 DLT 発現確率 ・ 用量除外規則適用に関する確率 ・ Single Patient Acceleration を採用する用量レベル ・ 用量増量減量基準（下限・上限） ・ 試験全体の最大被験者数 ・ 用量ごとの最大被験者数 ・ 最低用量での減量時の処理* ・ 乱数シード	数値 ・ 試験に組み入れられる総被験者数の期待値 ・ 試験の早期中止割合 ・ *最低用量で減量が選択された割合* ・ *最高用量で増量が選択された割合* ・ 各用量が MTD と選択される割合 ・ 各用量に組み入れられる被験者数の期待値 ・ 各用量に組み入れられる被験者の割合 図 ・ 真の DLT 発現確率のシナリオ ・ MTD 的中割合

*最低用量において、中止基準に抵触せずに最低用量で減量が支持された場合には試験を中止するか、或いは Retainment とみなして継続するか、状況に応じて確率的に考慮できるようにした

下線：BOIN デザイン特有の項目

斜線：3+3 デザイン特有の項目

図 2.2.10-1 ツール画面のキャプチャ

以下に本ツールに特徴的な機能について述べる。

結果テーブルのダウンロード機能

テーブル表示に DT パッケージ[5]を使用することで、結果を Excel 形式、CSV 形式等でダウンロード可能である。

```
shinyServer(function(input, output) {
    output$table_MTD <- DT::renderDT({
        DT::datatable(
            datatable_MTD(),
            colnames = c("Scenario", "% of selection (BOIN)", "Number of
patient treated (BOIN)", "% of patient treated (BOIN)", "% of selection
(3p3)", "Number of patient treated (3p3)", "% of patient treated (3p3)"
          ),
          rownames = FALSE,
          extensions = "Buttons",
          options = list(
            pageLength = length(scenario_list()),
            dom = "Bfrtip",
            buttons = c("colvis", "copy", "csv", "excel", "pdf", "print")
          )
        )
}) %>% bindEvent(input$Submit)
```

グラフのインタラクティブな表示

グラフ作成には ggplot2 パッケージ[6]を使用し、plotly パッケージ[7]の ggplotly()関数により描画している。これによりマウス操作による値の確認、拡大等が可能である。また、PNG 形式でグラフをダウンロードすることも可能である。

```
shinyServer(function(input, output) {
    dataplot_MTD <- reactive({
        selection_plot(datatable_MTD(),
        x_max = length(scenario_list()))
    }) %>% bindEvent(input$Submit)

    output$plot_MTD <- plotly::renderPlotly({
        dataplot_MTD() %>%
          plotly::ggplotly()
    })
})
```

※補助関数

- selection_plot()

  ggplot 関数によりグラフを作成する関数。

R スクリプトのダウンロード機能

modified BOIN デザイン動作特性の評価、modified 3+3 デザイン動作特性の評価、そ

の比較に関するそれぞれの処理は関数化して別ファイルに保持している。これらの関数スクリプトとシナリオを whisker パッケージ[8]により結合することで、ツールと同じ処理を行う R スクリプトを作成しダウンロードすることが可能である。
これにより結果の再現性を担保可能であり。ツールでは対応できないような、検討中の試験デザインに応じたコードの改変を容易に行うことができる。
以下にプログラムを示す。

① modified 3+3、modified BOIN、各用量を MTD と選択する割合を計算するコードと wisker テンプレート（詳細は[whisker テンプレート]参照）から、ダウンロードするスクリプトを生成する関数を返す。

```
generate_code <- function(
    scenario,
    ...,
    template = "templates/download_code.txt",
    file_BOIN_funcs = list("R/BOIN.R", "R/BOIN_OC.R"),
    file_3p3_funcs = list("R/OC_3p3.R"),
    file_selection_funcs = list("R/mtd_selection.R")) {
  function(file) {
    template <- paste(readLines(template), collapse = "¥n")
    f_BOIN <- paste(lapply(file_BOIN_funcs, read_source), collapse = "¥n"
    f_3p3 <- paste(lapply(file_3p3_funcs, read_source), collapse = "¥n")
    f_sel <- paste(lapply(file_selection_funcs, read_source), collapse = "
¥n")
    cat(
      file = file,
      append = FALSE,
      whisker::whisker.render(template, list(
        scenario_gen_code = data_preparation_code(scenario),
        functions_def_BOIN = f_BOIN,
        functions_def_3p3 = f_3p3,
        functions_selection = f_sel,
        ...
       ))
     )
  }
}
```

② ①で作成した関数を用いて、ダウンロードボタンをクリックした際にコードを生成しダウンロードする。

```
shinyServer(function(input, output) {
  output$downloadCodeButton <- renderUI({
    req(scenario())
    downloadButton("downloadCode", "Download Code")
  })

  output$downloadCode <- downloadHandler(
    filename = "boin_3p3_code.R",
    content = generate_code(
```

```
      scenario = scenario(), nsim = input$nsim, tDLT = input$tDLT,
      ptox = input$ptox, singlecht = input$singlecht,
      lambda1 = input$lambda1, lambda2 = input$lambda2,
      maxn = input$maxn, maxnDL = input$maxnDL, prob_sc = input$prob_sc,
      seed = input$seed
    )
  )
})
```

※補助関数

- read_source()
  ファイルから R コードをテキストとして読み込む関数。
- data_preparation_code()
  シナリオをまとめた CSV ファイルの情報を R コードに変換する関数。

[whisker テンプレート]

```
library(dplyr)
library(tidyr)
library(ggplot2)

{{{functions_def_BOIN}}}
{{{functions_def_3p3}}}
{{{functions_selection}}}
{{{scenario_gen_code}}}

result_BOIN <- BOIN_OC_Summary(
  scenario = scenario, tDLT = {{{tDLT}}}, nsim = {{{nsim}}},
  ptox = {{{ptox}}}, singlecht = {{{singlecht}}},
  lambda1 = {{{lambda1}}}, lambda2 = {{{lambda2}}}, maxn= {{{maxn}}},
  maxnDL = {{{maxnDL}}}, prob_sc = {{{prob_sc}}}, seed = {{{seed}}}
)

result_3p3 <- OC_3p3_Summary(
  scenario  = scenario, tDLT = {{{tDLT}}}, nsim = {{{nsim}}},
  singlecht = {{{singlecht}}}, seed = {{{seed}}}
)

selection <- selection_table(
  result_BOIN = result_BOIN, result_3p3 = result_3p3, tDLT = {{{tDLT}}}
)

selection_plot(data = selection, x_max = length(scenario))
```

3) 実行環境

Objects	Version
R	3.6.2
Shiny	1.7.4
BOIN	2.7.2

DT	0.18
ggplot2	3.3.3
plotly	4.9.3
whisker	0.4.1

参考文献

1) Liu S, Yuan Y. Bayesian optimal interval designs for phase I clinical trials. Journal of the Royal Statistical Society: Series C: Applied Statistics. 2015 Apr 1:507-23.

2) U.S. Food and Drug Administration .Drug Development Tools: Fit‐for‐Purpose Initiative.
https://www.fda.gov/drugs/development-approval-process-drugs/drug-development-tools-fit-purpose-initiative [Accessed 31 Jan 2024].

3) 30 日調査照会事項チェックリスト（抗悪性腫瘍剤分野）
https://www.pmda.go.jp/files/000252155.pdf [Published 27 Mar 2023].

4) Mi G, Bian Y, Wang X, Zhang W. SPA: Single patient acceleration in oncology dose-escalation trials. ContempClin Trials. 2021 Jun;105:106378.

5) DT
https://rstudio.github.io/DT/

6) ggplot2
https://ggplot2.tidyverse.org/

7) plotly
https://plotly.com/r/

8) whisker
https://github.com/edwindj/whisker

謝辞

本項の作成にあたりご助言及び資料提供いただきました、中外製薬の吉本拓矢様、中川雄貴様に感謝申し上げます。

3. まとめ

本文書では Python と R の活用事例を紹介した。本タスクフォースメンバーが収集した事例には、症例数の設計や高度なシミュレーション、データの可視化や業務の効率化に関わるものが多く、R の tidytlg パッケージの例のように総括報告書や CTD の作成のための直接的な利用の事例は少なかった。この背景には、前述の OSS 利用状況アンケートの報告書[1)]の R の利用状況の結果にもあるように、日本の製薬業界では、総括報告書や CTD のための帳票・データセット作成業務よりも、その前後の業務や社内資料の作成のために OSS の利用が先行していることがあると考えられる。しかし、それらの業務の中には臨床試験デザインの策定のように重要な業務に関わるものもあり、日本の製薬業界でも OSS が既に有効利用されていると考えて良いのではないか。

一方、R で顕著であるが、総括報告書や CTD のための帳票・データセット作成業務に直接的に OSS を利用する取り組みも海外では進んでいる。その一つの取り組みは pharmaverse[2)]であり、tidytlg パッケージをはじめとする帳票・データセット作成に利用できるパッケージの開発や情報共有を有志が行っている。また、Novo Nordisk 社は、総括報告書や CTD のための帳票・データセット作成業務に R を利用し、FDA への承認申請を行った[3)]。この際、R のプログラムも FDA へ提出され、何度か FDA から問い合わせを受けたものの、提出した R プログラムを利用して FDA の審査は行われているとのことである。

しかし、OSS 利用状況アンケートの報告書[1)]の結果にもあるように、「OSS の CSV・信頼性」の懸念が日本の製薬業界に依然としてある。一方、海外の製薬会社の OSS の信頼性担保の事例であれば、例えば 2023 年に神戸で行われた PHUSE の Single Day Event[4)]で幾つか紹介されていた。OSS を総括報告書や CTD のために利用するのであれば、日本の製薬業界でも、OSS の信頼性の担保として何をすべきかを具体的に理解していくことが必要だと考えられる。

1) OSS 利用状況アンケート報告書
https://www.jpma.or.jp/information/evaluation/results/allotment/g75una0000001axl-att/DS_202204_oss_questionaire.pdf

2) pharmaverse
https://pharmaverse.org/

3) Learnings from the First R-Based Submission to FDA by Novo Nordisk
https://www.r-bloggers.com/2024/01/learnings-from-the-first-r-based-submission-to-fda-

by-novo-nordisk/

4) PHUSE Kobe, Japan SDE 2023
https://www.phuse-events.org/attend/frontend/reg/thome.csp?pageID=26389&eventID=42&traceRedir=2

日本製薬工業協会 医薬品評価委員会 データサイエンス部会 2023 年度タスクフォース 5
オープンソースソフトウェアの活用と留意事項チーム

加藤　智子*　　サノフィ㈱
堀田　真一**　　住友ファーマ㈱
坂上　拓**　　中外製薬㈱
新井　賢太郎　　ノバルティス ファーマ㈱（～2023 年 12 月）
松永　友貴　　ノバルティス ファーマ㈱
鈴木　一平　　エーザイ㈱
東島　正堅　　イーライリリー㈱
西村　力丸　　ヤンセンファーマ㈱
生井　伴幸　　中外製薬㈱
保田　昂之　　中外製薬㈱

*担当副部会長
**推進委員