

# 10 個のデータファイルを 1 ファイルに統合する方法 by R

Hiroya Sahashi

2019 年 6 月 4 日

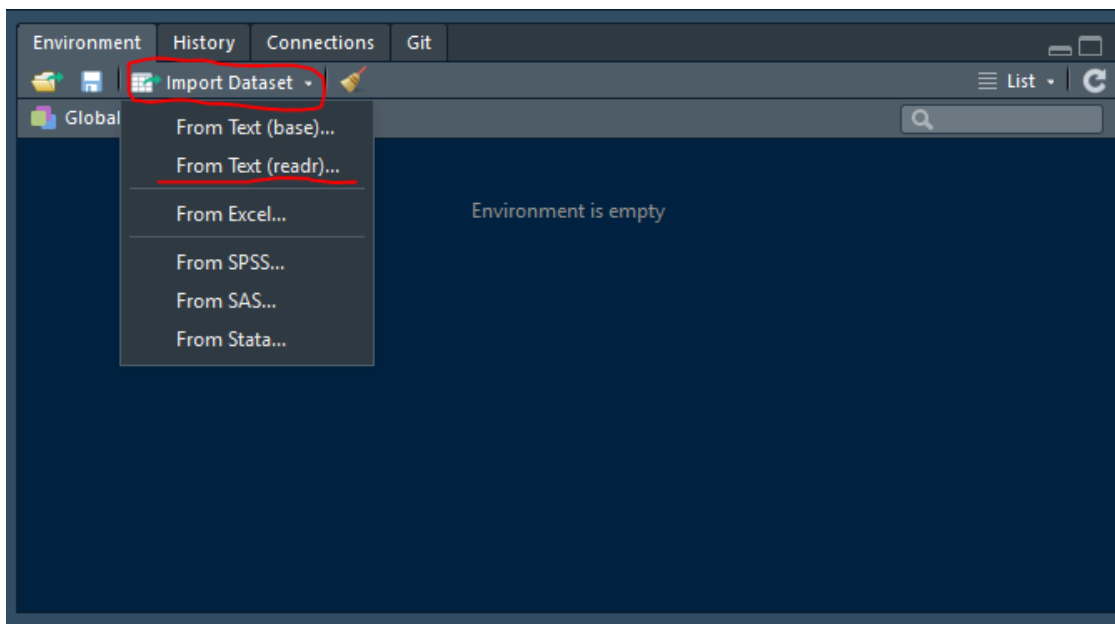
## 解説すること

この文書では、あるフォルダにある複数のデータファイル(csv ファイルなど) を一つのファイルに統合して保存する方法を解説します。

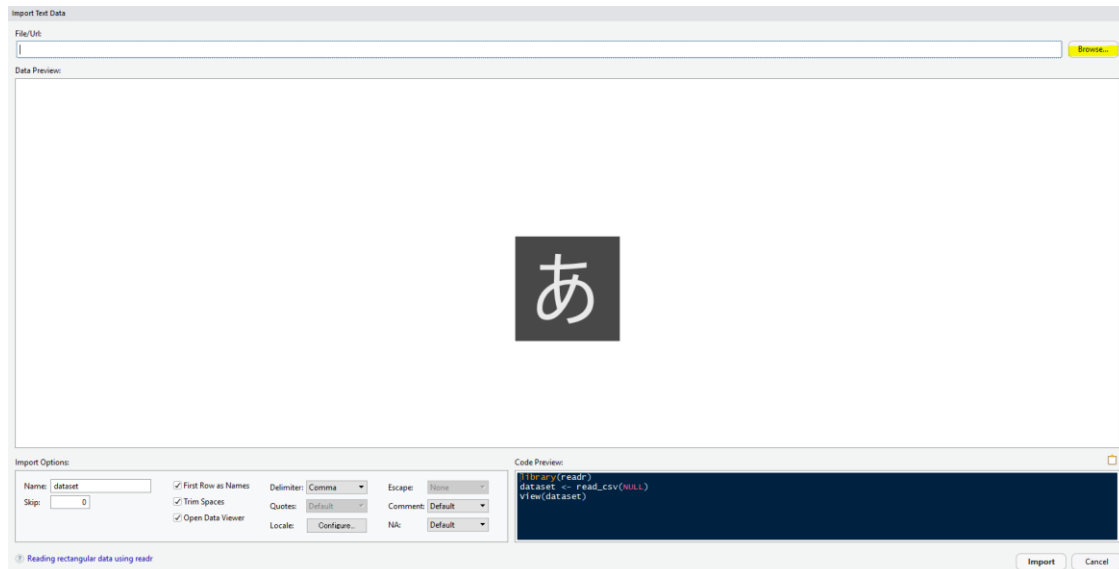
## データの確認

Rstudio には UI (ユーザーインターフェース) でデータを確認する機能があるので、その機能を利用してみましょう。

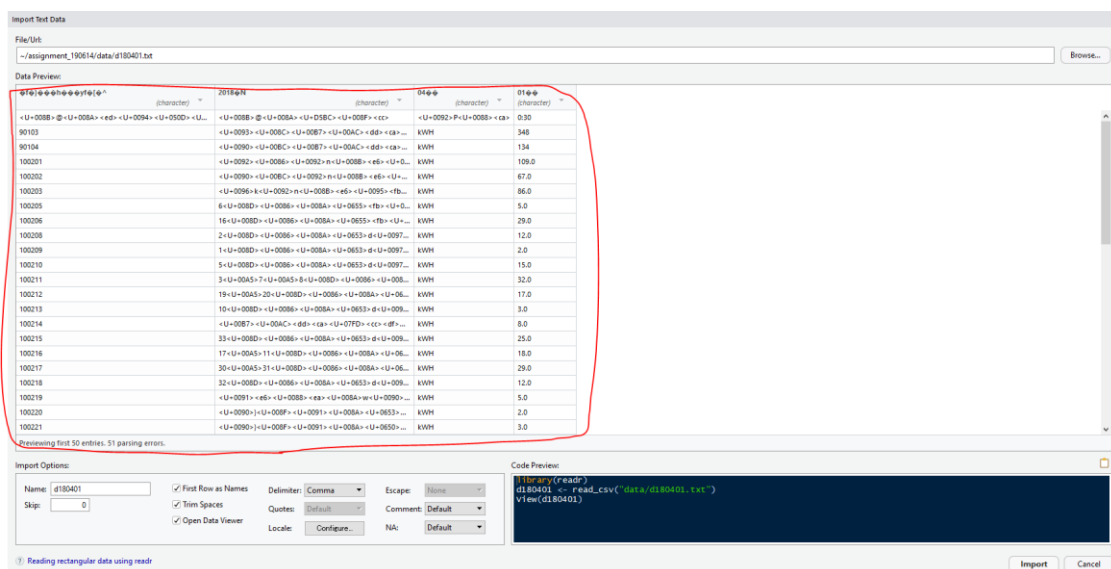
まず右上の **Workspace** パネルにある *import Dataset* ボタンをクリックし、*From Text(readr)*の部分をクリックしてください。R から *readr* パッケージをインストールするよう求められた場合は、**Yes** を選んでください。



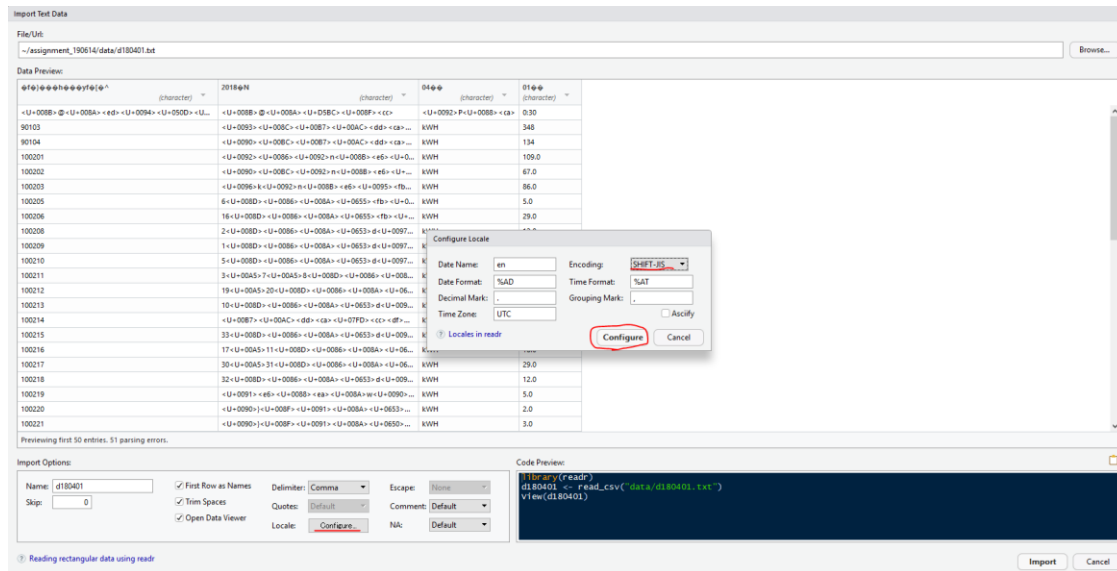
すると、このような画面が立ち上がるので、**Browse** ボタンをクリックし、  
~/assignment\_190614/data にある d180401.txt を開いてみてください。



d180401.txt の Data Preview を確認すると、列名やその値が正確に読み込めていない（文字化けが生じている）ことが分かります。これは、文字符号化（エンコード）形式が誤っているためです。日本語の PC 環境は主に **UTF-8** あるいは **Shift-JIS** が文字符号化形式が利用されているため、最低限この 2 種類は覚えておいてください。スマート BEMS から得られるデータの文字符号化形式は **Shift-JIS** です。



文字符号化形式を変更するには、Locale:の *Configure...* ボタンをクリックし、ポップアップとして表示されるウィンドウの Encoding を *Shift-JIS* に変更し、ポップアップウィンドウ上の *Configure* をクリックしてください。



文字符号化形式を *Shift-JIS* に設定(configure)することで、Data Preview の文字化けが解消されます。そして、*Import* ボタンをクリックすることで、d18041 というオブジェクト名の *data.frame* 型のデータとして対象とする csv ファイルを読み込むことができます。ちなみに、この作業は R の *Console* に *Code Preview* に表示されたコードを実行したことに過ぎません。但し、ファイルの中身がよく分からないときなどには、この *Import Dataset* の UI は便利な機能なので活用するようにしてください。

Import Text Data

File/Url:  
~/assignment\_190814/data/d180401.txt

Data Preview:

デモン データ (character)	2018年 (character)	04月 (character)	01日 (character)
読者番号	読者名	単位	0.30
90103	東京電力	KWH	340
90104	関西電力	KWH	194
100201	中部電力	KWH	109.0
100202	西地区電力	KWH	67.0
100203	北地区電力	KWH	86.0
100205	6号電力	KWH	5.0
100206	10号電力	KWH	29.0
100208	2号電力	KWH	12.0
100209	1号電力	KWH	2.0
100210	5号電力	KWH	15.0
100211	3-7-8号電力	KWH	32.0
100212	19-20号電力	KWH	17.0
100213	10号電力	KWH	3.0
100214	4号電力	KWH	8.0
100215	11号電力	KWH	25.0
100216	17-19号電力	KWH	18.0
100217	30-31号電力	KWH	29.0
100218	12号電力	KWH	12.0
100219	第一号電力	KWH	5.0
100220	回線電力	KWH	2.0

Previewing first 50 entries. 51 parsing errors.

Import Options:

Name: d180401  
Skip: 0  
☒ First Row as Names  
☒ Trim Spaces  
☒ Open Data Viewer  
Delimiter: Comma  
Quotes: Default  
Locale: Configure...  
Escape: None  
Comment: Default  
NA: Default

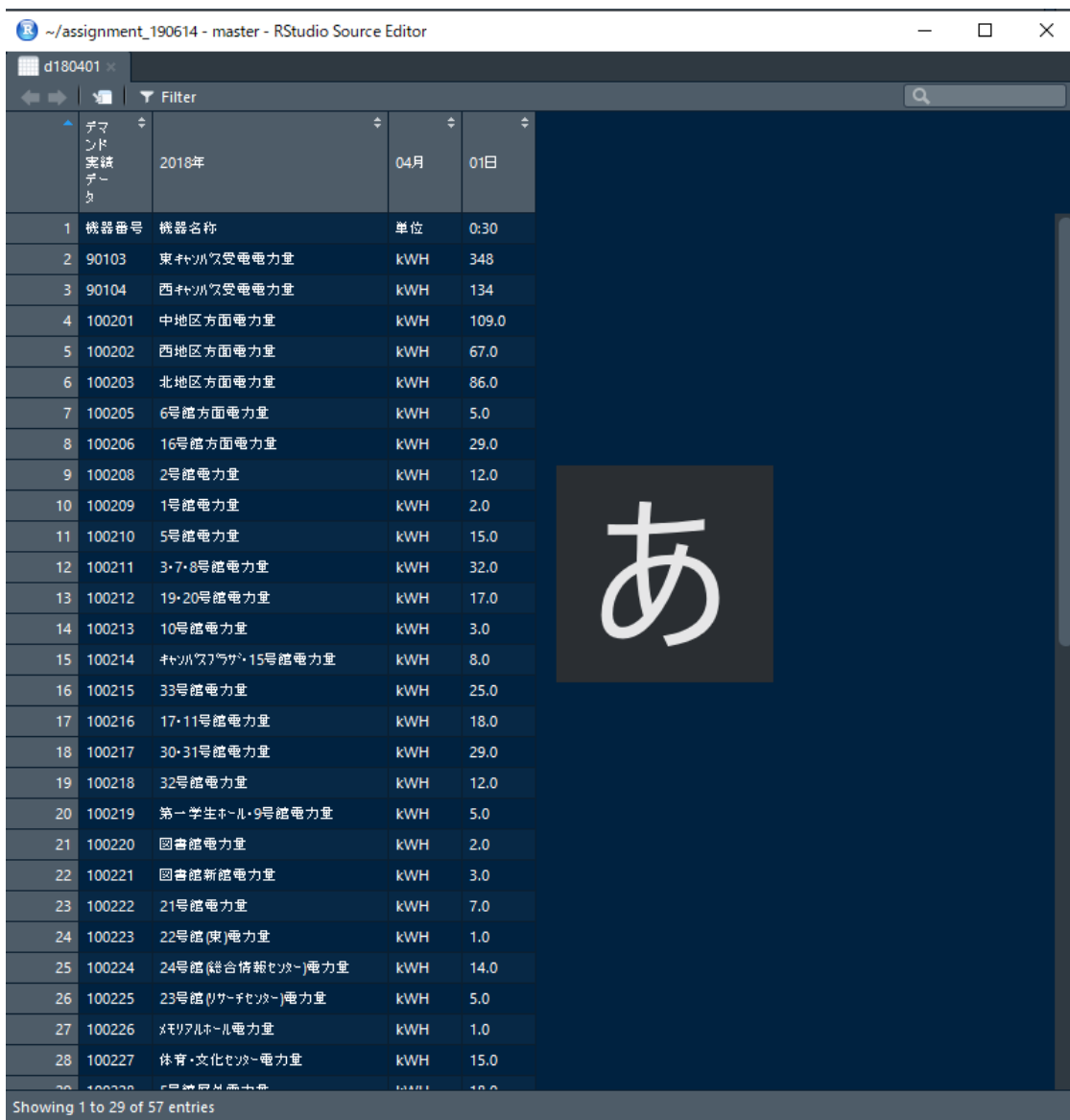
Code Preview:

```
library(readr)
d180401 <- read_csv("data/d180401.txt", locale = locale(encoding = "SHIFT-JIS"))
view(d180401)
```

Reading rectangular data using readr

Import Cancel

ただ、読み込んだオブジェクト `d180401` に違和感を覚えないでしょうか？



~/assignment\_190614 - master - RStudio Source Editor

d180401

Filter

	デマ ンド 実績 デー タ	2018年	04月	01日
1	機器番号	機器名称	単位	0:30
2	90103	東キャンパス受電電力量	kWH	348
3	90104	西キャンパス受電電力量	kWH	134
4	100201	中地区方面電力量	kWH	109.0
5	100202	西地区方面電力量	kWH	67.0
6	100203	北地区方面電力量	kWH	86.0
7	100205	6号館方面電力量	kWH	5.0
8	100206	16号館方面電力量	kWH	29.0
9	100208	2号館電力量	kWH	12.0
10	100209	1号館電力量	kWH	2.0
11	100210	5号館電力量	kWH	15.0
12	100211	3・7・8号館電力量	kWH	32.0
13	100212	19・20号館電力量	kWH	17.0
14	100213	10号館電力量	kWH	3.0
15	100214	キャンパスラゲ・15号館電力量	kWH	8.0
16	100215	33号館電力量	kWH	25.0
17	100216	17・11号館電力量	kWH	18.0
18	100217	30・31号館電力量	kWH	29.0
19	100218	32号館電力量	kWH	12.0
20	100219	第一学生ホール・9号館電力量	kWH	5.0
21	100220	図書館電力量	kWH	2.0
22	100221	図書館新館電力量	kWH	3.0
23	100222	21号館電力量	kWH	7.0
24	100223	22号館(東)電力量	kWH	1.0
25	100224	24号館(総合情報センサ-)電力量	kWH	14.0
26	100225	23号館(リサーチセンサ-)電力量	kWH	5.0
27	100226	メモリアルホール電力量	kWH	1.0
28	100227	体育・文化センサ-電力量	kWH	15.0
29	100228	5号館旧館電力量	kWH	10.0

Showing 1 to 29 of 57 entries

- あれ、時刻が 0 時半のデータしかないぞ？と思えた人はセンスがいいです。

R で読み込むデータは列名に依存し、列名が無いと判断される場合、その下の行方向のデータは読み取られません。つまり、0:30 を含む行を列名とする方が、良いと考えられるわけです。

では、もう一度 *import Dataset* ボタンをクリックし、先程と同様な設定（文字符号化形式 Shift-JIS）にしてください。そして、追加設定として *Skip* のテキストボックスの数字を 0 から 1 に変更し、再度 *Import* ボタンを押してください。データの 1 行目を無視して、データが読み込まれます。

Import Text Data

File/Url:  
~/assignment\_190614/data/d180401.txt

Data Preview:

観測番号 (Integer)	観測名称 (Character)	単位 (Character)	0:30 (Double)	1:00 (Double)	1:30 (Double)	2:00 (Double)	2:30 (Double)	3:00 (Double)	3:30 (Double)	4:00 (Double)	4:30 (Double)	5:00 (Double)	5:30 (Double)	6:00 (Double)	6:30 (Double)	7:00 (Double)	7:30 (Double)	8:00 (Double)
90103	東ヶ丘7号観測電力量	KWH	348	348	351	350	348	352	360	370	358	360	343	343	348	333	339	328
90104	西ヶ丘7号観測電力量	KWH	134	134	136	132	134	132	132	131	133	131	131	130	141	138	151	151
100201	中野区水産電力量	KWH	109	112	114	115	109	119	117	122	119	115	113	120	125	120	126	126
100202	西野区水産電力量	KWH	67	66	66	67	66	67	74	77	76	76	67	66	51	47	43	36
100203	北野区水産電力量	KWH	86	86	86	85	87	86	85	84	86	85	87	86	87	87	88	86
100205	6号観測電力量	KWH	5	5	5	5	4	5	5	4	5	5	5	5	4	5	4	4
100206	16号観測電力量	KWH	29	30	30	29	29	30	29	30	29	29	30	29	30	30	29	30
100208	2号観測電力量	KWH	12	12	13	11	12	11	15	22	18	11	12	12	15	13	24	26
100209	1号観測電力量	KWH	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
100210	5号観測電力量	KWH	15	14	15	14	14	19	14	15	14	15	14	15	19	16	14	16
100211	3号観測電力量	KWH	32	36	37	36	36	36	37	35	37	37	36	36	37	38	36	38
100212	19-20号観測電力量	KWH	17	17	19	18	17	19	18	17	19	18	20	22	18	18	17	17
100213	10号観測電力量	KWH	3	4	4	4	4	4	3	4	4	4	4	4	4	3	4	3
100214	4号観測電力量	KWH	8	7	8	9	7	8	7	8	7	7	7	8	9	9	9	9
100215	33号観測電力量	KWH	25	25	26	26	26	27	25	26	25	26	26	26	26	26	27	27
100216	17-18号観測電力量	KWH	18	18	18	18	18	19	17	18	18	17	19	18	17	18	16	16
100217	30-31号観測電力量	KWH	29	31	30	30	30	30	29	30	30	31	30	31	30	31	31	31
100218	32号観測電力量	KWH	12	12	12	12	11	12	11	12	12	11	12	11	13	12	13	13
100219	第一号観測電力量	KWH	5	6	5	6	6	6	5	6	6	5	6	6	5	6	6	6
100220	回線観測電力量	KWH	2	2	2	3	2	2	3	2	2	3	2	2	3	2	2	2

Import Options:

Name: d180401  
Skip: 1  
First Row as Names: ☒  
Trim Spaces: ☒  
Open Data Viewer: ☒  
Delimiter: Comma  
Quotes: Default  
Escape: None  
Comment: Default  
Locale: Configure  
NA: Default

Code Preview:

```
library(readr)
d180401 <- read_csv("data/d180401.txt", locale = locale(encoding = "SHIFT_JIS"),
  skip = 1)
view(d180401)
```

Reading rectangular data using readr

Import Cancel

Environment History Connections Git

Import Dataset

Global Environment

Data

d180401 56 obs. of 55 variables

最終的に Workspace パネルに *d180401 56 Obs. of 55 variables*(56 行 55 変数)と表示  
されていれば、大丈夫です。

## for ループと if 分岐の R での書き方

プログラミングを少しかじったことがある人なら、for ループや if 分岐は使った  
ことがあると思います。そこで、ループ変数 *i* が 1~10 の範囲(python 等の 0 から始  
まるリストインデックスと異なり、1 から始まるため) で、偶数か奇数かを判定  
するプログラムを考えてみようと思います。

```

for (i in 1:10) {
  if(i%2 == 0) { # i を 2 で割った時の剰余が 0 : 偶数
    print(paste0(i, "は偶数です"))
  } else { # それ以外 i を 2 で割った時の剰余が 0 ではない : 奇数
    print(paste0(i, "は奇数です"))
  }
}

## [1] "1 は奇数です"
## [1] "2 は偶数です"
## [1] "3 は奇数です"
## [1] "4 は偶数です"
## [1] "5 は奇数です"
## [1] "6 は偶数です"
## [1] "7 は奇数です"
## [1] "8 は偶数です"
## [1] "9 は奇数です"
## [1] "10 は偶数です"

```

上記のコードは、初歩的な for ループの書き方です。例えば、`for (i in 1:10) {print("Hello")}` というコードであれば、10 回コンソール上に“Hello”と改行入りで表示されます。for ループ内で if と [比較演算子](#) を用いた論理式を用いることで、偶数回目は A の命令、奇数回目は B の命令というように条件分岐させることができます(if 分岐)。なお、`paste0` 関数は引数(arguments)として指定した文字列オブジェクトをスペースなしに連結することができる関数です。[計算演算子](#)もできるだけ覚えるようにしてください。

## 目標とする year2018.csv ファイルの仕様について

1. 1 列目に時刻日付ラベル (format:“%Y-%m-%d %H:%M:%S”, tz:“UTC”)、2 列目以降は数値(numeric)型のデータが格納された data.frame
2. 行（下）方向に時が進む時系列データであり、時間間隔は 1 時間とする
3. 文字符号化形式 UTF-8 の csv ファイルとして保存する

作成する year2018.csv ファイルはこの仕様に従う必要があります。

## 具体的な R スクリプトファイル(.R 拡張子)の書き方

### 外部パッケージの読み込み

```
library(tidyverse)

## -- Attaching packages -----
----- tidyverse 1.2.1 --
## √ ggplot2 3.0.0      √ purrr   0.2.5
## √ tibble  1.4.2      √ dplyr  0.7.7
## √ tidyr   0.8.1      √ stringr 1.3.1
## √ readr   1.1.1      √ forcats 0.3.0

## -- Conflicts -----
----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

R パッケージの **tidyverse** はデータの加工に適した複数のパッケージを作業環境下に付与してくれるパッケージです。この場合、**ggplot2** や **dplyr** などの 8 種類のパッケージが作業環境下に付与され、**filter** 関数(**dplyr** と **stats**)と **lag** 関数(**dplyr** と **stats**)の関数名の干渉が起こっています。**tidyverse** パッケージは頻繁に利用するパッケージなので、使用したい外部パッケージが定まっていないときは、とりあえず読み込んでしまってもよいでしょう w

### 作業環境の設定

R プログラムが実行されているフォルダのことをワーキング（カレント）ディレクトリと呼びます。まず、日毎のデータが保存してある **data** フォルダをワーキングディレクトリに設定しましょう。

```
# ワーキングディレクトリを相対パスで指定
work_dir <- "~/\\assignment_190614\\data\\"
# ワーキングディレクトリを work_dir に設定
setwd(work_dir)
# ワーキングディレクトリの確認
getwd()

## [1] "C:/Users/Sahashi/Documents/assignment_190614/data"
```



ワーキングディレクトリを確認するコマンド `getwd()` を実行し、“C:/Users/ユーザー名/Documents/assignment\_190614/data” という文字列が返されれば、適切に設定ができています。

プログラマは、同様な単調な操作を書き下すことを嫌います。今回の課題はワーキングディレクトリ内のファイルに同様な処理を施していくことが予想されます。そのようなときに、`for` ループを利用します。今回は `dir` 関数を利用して、ワーキングディレクトリ内のファイル名をベクトル型で取得します。

```
file_names <- dir(work_dir)
file_names

## [1] "d180401.txt" "d180402.txt" "d180403.txt" "d180404.txt" "d180405.
## [6] "d180406.txt" "d180407.txt" "d180408.txt" "d180409.txt" "d180410.
txt"
```

## 時刻ラベルの定義

時刻ラベルについては、例えば 0 時から 0 時半までに記録された電力積算値の時刻ラベルを“00:00:00”としてラベリングすることにします。データ分析においては、観測値の時刻ラベルが記録開始時刻なのか、記録終了時刻なのかという細かい点にも注意を払うようにしてください。今回は、記録開始時刻をラベリングするデータとします。30 分間隔なので 1 日当たり 48 個の時刻ラベルが付与されることになります。

```
# 時刻ラベルの定義
hour_lab <- c("00:00:00", "00:30:00", "01:00:00", "01:30:00", "02:00:00",
"02:30:00",
"03:00:00", "03:30:00", "04:00:00", "04:30:00", "05:00:00",
"05:30:00",
"06:00:00", "06:30:00", "07:00:00", "07:30:00", "08:00:00",
"08:30:00",
"09:00:00", "09:30:00", "10:00:00", "10:30:00", "11:00:00",
"11:30:00",
"12:00:00", "12:30:00", "13:00:00", "13:30:00", "14:00:00",
"14:30:00",
"15:00:00", "15:30:00", "16:00:00", "16:30:00", "17:00:00",
"17:30:00",
"18:00:00", "18:30:00", "19:00:00", "19:30:00", "20:00:00",
"20:30:00",
"21:00:00", "21:30:00", "22:00:00", "22:30:00", "23:00:00",
"23:30:00" )
```

## for ループブロックの大枠

以下に for ループブロックの大枠を示します。

```
for (i in 1:length(file_names)) {  
  
}
```

length 関数をベクトルに適用することで、そのベクトル内にいくつの要素があるかが分かります。ベクトル型のオブジェクト `file_names` の範囲は 1:10 であるので 10 が返されます。従って、この for ループはループ変数 `i` が 1 ずつ増加（初回は除く）しながら、10 回の処理を行うということです。

## i == 1(1 回目の処理)について

```
i <- 1  
# ファイルから日付の取得  
temp_date <- read_csv(paste0(work_dir, file_names[i]), col_names = F,  
                      locale = locale(encoding = "SJIS"),  
                      skip = 0, n_max = 1)  
  
## Parsed with column specification:  
## cols(  
##   X1 = col_character(),  
##   X2 = col_character(),  
##   X3 = col_character(),  
##   X4 = col_character()  
## )  
  
# 日付ラベルの作成  
date_lab <- paste0(substr(temp_date$X2, 1, 4), "-",  
                  substr(temp_date$X3, 1, 2), "-",  
                  substr(temp_date$X4, 1, 2))  
  
# データを取得  
temp <- read_csv(paste0(work_dir, file_names[i]),  
                locale = locale(encoding = "SJIS"), col_names = F,  
                skip = 2)  
  
## Parsed with column specification:  
## cols(  
##   .default = col_double(),  
##   X1 = col_integer(),  
##   X2 = col_character(),  
##   X3 = col_character()  
## )
```

```
## See spec(...) for full column specifications.

# 転置行列化, 最小値・最大値・平均値・合計値の項目を除く
# 要素化を防ぐため、数値データのみとする
t_temp <- as.data.frame(t(dplyr::select(temp, X4:X51)), row.names = F)

# 一日ごとの日付時刻ラベル作成
date_time <- paste(date_lab, hour_lab, sep = " ")

n_temp <- t_temp
n_lab <- date_time
```

1 回目に定義した `n_temp`(各系列のデータ部分)と `n_lab` (日付時刻ラベル) を `rbind` 関数を用いてデータ下部に結合させていきます。

## i > 1(else)の処理について

```
i <- 2
# ファイルから日付の取得
temp_date <- read_csv(paste0(work_dir, file_names[i]), col_names = F,
                      locale = locale(encoding = "SJIS"),
                      skip = 0, n_max = 1)

## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_character(),
##   X4 = col_character()
## )

# 日付ラベルの作成
date_lab <- paste0(substr(temp_date$X2, 1, 4), "-",
                  substr(temp_date$X3, 1, 2), "-",
                  substr(temp_date$X4, 1, 2))

# データを取得
temp <- read_csv(paste0(work_dir, file_names[i]),
                 locale = locale(encoding = "SJIS"), col_names = F, skip
                 = 2)

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_integer(),
##   X2 = col_character(),
```

```

## X3 = col_character()
## )

## See spec(...) for full column specifications.

# 列名のための文字列を取得
cnames <- temp$X2

# 転置行列化, 最小値・最大値・平均値・合計値の項目を除く
# 要素化を防ぐため、数値データのみとする
t_temp <- as.data.frame(t(dplyr::select(temp, X4:X51)), row.names = F)

# 列名の追加
names(t_temp) <- cnames
names(n_temp) <- cnames

# 列名で整列させる(元データに余分な全角スペースが入っている)
c1 <- t_temp$`東キャンパス受電電力量`
c2 <- t_temp$`西キャンパス受電電力量`
c3 <- t_temp$`中地区方面電力量`
c4 <- t_temp$`西地区方面電力量`
c5 <- t_temp$`北地区方面電力量`
c6 <- t_temp$`6号館方面電力量`
c7 <- t_temp$`16号館方面電力量`
c8 <- t_temp$`2号館電力量`
c9 <- t_temp$`1号館電力量`
c10 <- t_temp$`5号館電力量`
c11 <- t_temp$`3・7・8号館電力量`
c12 <- t_temp$`19・20号館電力量`
c13 <- t_temp$`10号館電力量`
c14 <- t_temp$`キャンパスプラザ・15号館電力量`
c15 <- t_temp$`33号館電力量`
c16 <- t_temp$`17・11号館電力量`
c17 <- t_temp$`30・31号館電力量`
c18 <- t_temp$`32号館電力量`
c19 <- t_temp$`第一学生ホール・9号館電力量`
c20 <- t_temp$`図書館電力量`
c21 <- t_temp$`図書館新館電力量`
c22 <- t_temp$`21号館電力量`
c23 <- t_temp$`22号館(東)電力量`
c24 <- t_temp$`24号館(総合情報センター)電力量`
c25 <- t_temp$`23号館(リサーチセンター)電力量`
c26 <- t_temp$`メモリアルホール電力量`

```

```

c27 <- t_temp$` 体育・文化センター電力量`
c28 <- t_temp$` 5 号館屋外電力量`
c29 <- t_temp$` 図書館(空調)電力量`
c30 <- t_temp$` 22 号館(西)電力量`
c31 <- t_temp$` 25 号館方面電力量`
c32 <- t_temp$` グラウンド 方面電力量`
c33 <- t_temp$` 70 号館方面電力量`
c34 <- t_temp$` 50 号館方面電力量`
c35 <- t_temp$` 50・51 号館電力量`
c36 <- t_temp$` 53 号館(地上)電力量`
c37 <- t_temp$` 52 号館電力量`
c38 <- t_temp$` フロンティア研究棟電力量`
c39 <- t_temp$` 高電圧実験室電力量`
c40 <- t_temp$` 体育館電力量`
c41 <- t_temp$` グラウンド 電力量`
c42 <- t_temp$` 29 号館(クラブ・サークルプラザ)電力量`
c43 <- t_temp$` テニスコート電力量`
c44 <- t_temp$` 53 号館(屋上)電力量`
c45 <- t_temp$` 超伝導実験棟電力量`
c46 <- t_temp$` 72 号館電力量`
c47 <- t_temp$` 体育館(空調)電力量`
c48 <- t_temp$` 55 号館電力量`
c49 <- t_temp$` 70・71 号館電力量`
c50 <- t_temp$` 武道体育館電力量`
c51 <- t_temp$` 25 号館電力量`
c52 <- t_temp$` 14 号館方面電力量`
c53 <- t_temp$` その他電力量`
c54 <- t_temp$` 不言実行館電力量`
c55 <- t_temp$` 外気温度`
c56 <- t_temp$` 外気湿度`

```

*# この順に強制的に並べ替え*

```

t_temp2 <- cbind.data.frame(c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,
                             c11,c12,c13,c14,c15,c16,c17,c18,c19,c20,
                             c21,c22,c23,c24,c25,c26,c27,c28,c29,c30,
                             c31,c32,c33,c34,c35,c36,c37,c38,c39,c40,
                             c41,c42,c43,c44,c45,c46,c47,c48,c49,c50,
                             c51,c52,c53,c54,c55,c56)

```

*# 一日ごとの日付時刻ラベル作成*

```

date_time <- paste0(date_lab, " ", hour_lab)

```

```

x_temp <- t_temp2
names(x_temp) <- cnames

```

```
# n_temp の更新
n_temp <- rbind(n_temp, x_temp)
x_lab <- date_time
# n_lab の更新
n_lab <- cbind(n_lab, x_lab)
```

i が 2 以上(つまり、else のとき)は上記のコードの処理を行います。

## for ループ後の処理について

for ループ後は念のため、work\_dir 内のファイル数分ループするか判定(judge)を行います。

```
# データ結合の判定
judge <-
  if(i == length(file_names)) {
    print("結合成功")
  } else {
    print("結合失敗")
  }
```

```
## [1] "結合失敗"
```

この文書のように“結合失敗”と表示される場合は、for ループがうまくいかなかった（ファイル数分ループしなかった）ということなので、for ループブロック内のコードを見直す必要があります。

“結合成功”と表示される場合は、以下のコードをコピーすれば 1 時間間隔の指定した仕様のデータが完成するはずです。

```
# 文字列の余分な全角スペースの削除
cnames2 <- gsub(" ", "", cnames, fixed = T)

# 列名の追加
names(n_temp) <- cnames2
# 型を数値型(numeric)に統一
y <- n_temp %>% mutate_if(is.character, as.numeric)

# ラベルと結合
label <- c(n_lab)
dataset <- cbind(label, y) %>% as_data_frame()

# 1 時間間隔のデータに変換
year2018 <- dataset %>% mutate(
  date_time = paste0(substr(label, 1, 14), "00:00")
```

```
) %>%  
  group_by(date_time) %>%  
  select(-label) %>%  
  summarise_each(funs(sum))  
  
## `summarise_each()` is deprecated.  
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.  
## To map `funs` over all variables, use `summarise_all()`
```

## データの保存

最後にデータの保存です。文字符号化形式 UTF-8 の csv ファイルとして保存できる `write_excel_csv` という便利な関数があるので、この関数を用いて `year2018` オブジェクトを `'year2018.csv'` として保存してください。

```
# save 用のディレクトリ  
save_dir <- "~\\assignment_190614\\"  
# UTF-8 の csv ファイルとして保存 (ここでは実行しない)  
# write_excel_csv(year2018, file.path(save_dir, "year2018.csv"))
```

## 最後に

ここまでの内容をまとめた R Script ファイル(`combineData_example.R`)を作成してあるので、各自の PC 環境でその挙動を確かめてみてください。R Script の実行は、RStudio の Source editor 右上部の *Source* ボタンをクリックするだけです。下の 2 番目の図のようなデータセットが出来上がるはずです。

combineData\_example.R | commentary doc.Rmd | Source

```
1 # 外部パッケージの読み込み -----
2 library(tidyverse)
3
4 # ディレクトリの設定 -----
5 # 相対パスでwork_dirを指定
6 work_dir <- "\\assignment_190614\\data\\"
7
8 # 時刻ラベルの定義 -----
9
10 hour_lab <- c("00:00:00", "00:30:00", "01:00:00", "01:30:00", "02:00:00", "02:30:00",
11              "03:00:00", "03:30:00", "04:00:00", "04:30:00", "05:00:00", "05:30:00",
12              "06:00:00", "06:30:00", "07:00:00", "07:30:00", "08:00:00", "08:30:00",
13              "09:00:00", "09:30:00", "10:00:00", "10:30:00", "11:00:00", "11:30:00",
14              "12:00:00", "12:30:00", "13:00:00", "13:30:00", "14:00:00", "14:30:00",
15              "15:00:00", "15:30:00", "16:00:00", "16:30:00", "17:00:00", "17:30:00",
16              "18:00:00", "18:30:00", "19:00:00", "19:30:00", "20:00:00", "20:30:00",
17              "21:00:00", "21:30:00", "22:00:00", "22:30:00", "23:00:00", "23:30:00")
18
19
20
21
22 # Forループ処理 -----
23 for (i in 1:length(File_names)) {
24   if(i == 1) {
25     # フォルダから日付の取得
26     temp_date <- read_csv(paste0(work_dir, file_names[i]), col_names = F,
27                           locale = locale(encoding = "SJIS"),
28                           skip = 0, n_max = 1)
29
30     # 日付ラベルの作成
31     date_lab <- paste0(substr(temp_date$x2, 1, 4), "-",
32                        substr(temp_date$x3, 1, 2), "-",
33                        substr(temp_date$x4, 1, 2))
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867

```

最後らへんに大幅に解説をすっ飛ばした部分もありますが、以上で「10 個のデータファイルを 1 ファイルに統合する方法 by R」の解説を終わります。

R 言語に限らずプログラミングにおいて、予期せぬエラーが生じることは日常茶飯事です。分からないことがあれば、気軽に佐橋に相談してください！