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

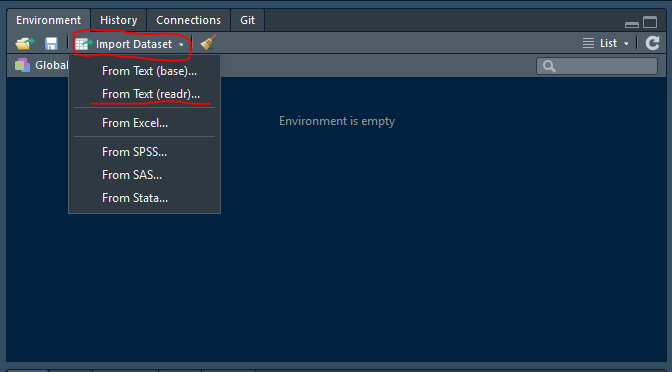
# 解説すること

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。

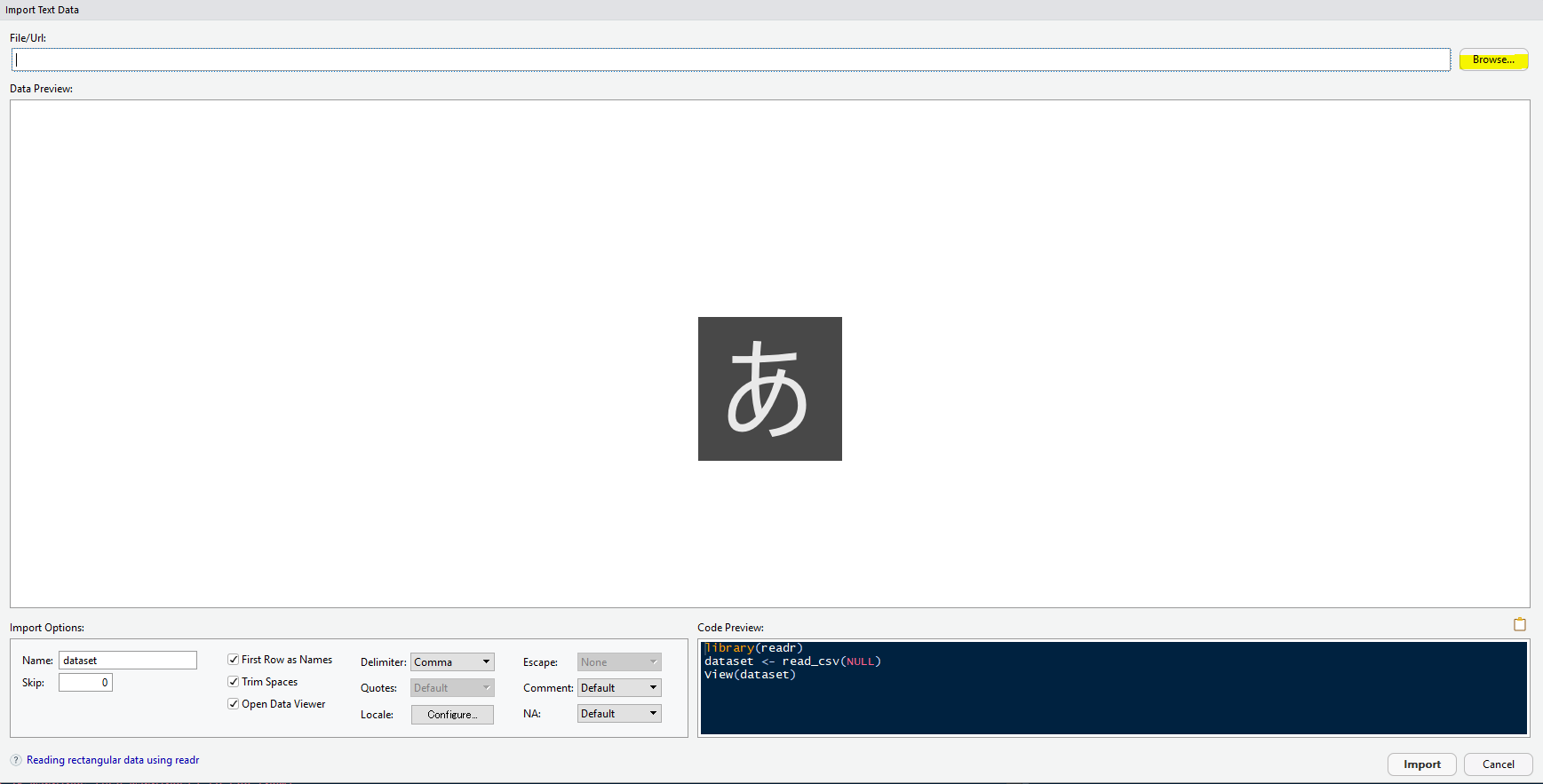
# データの確認

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。

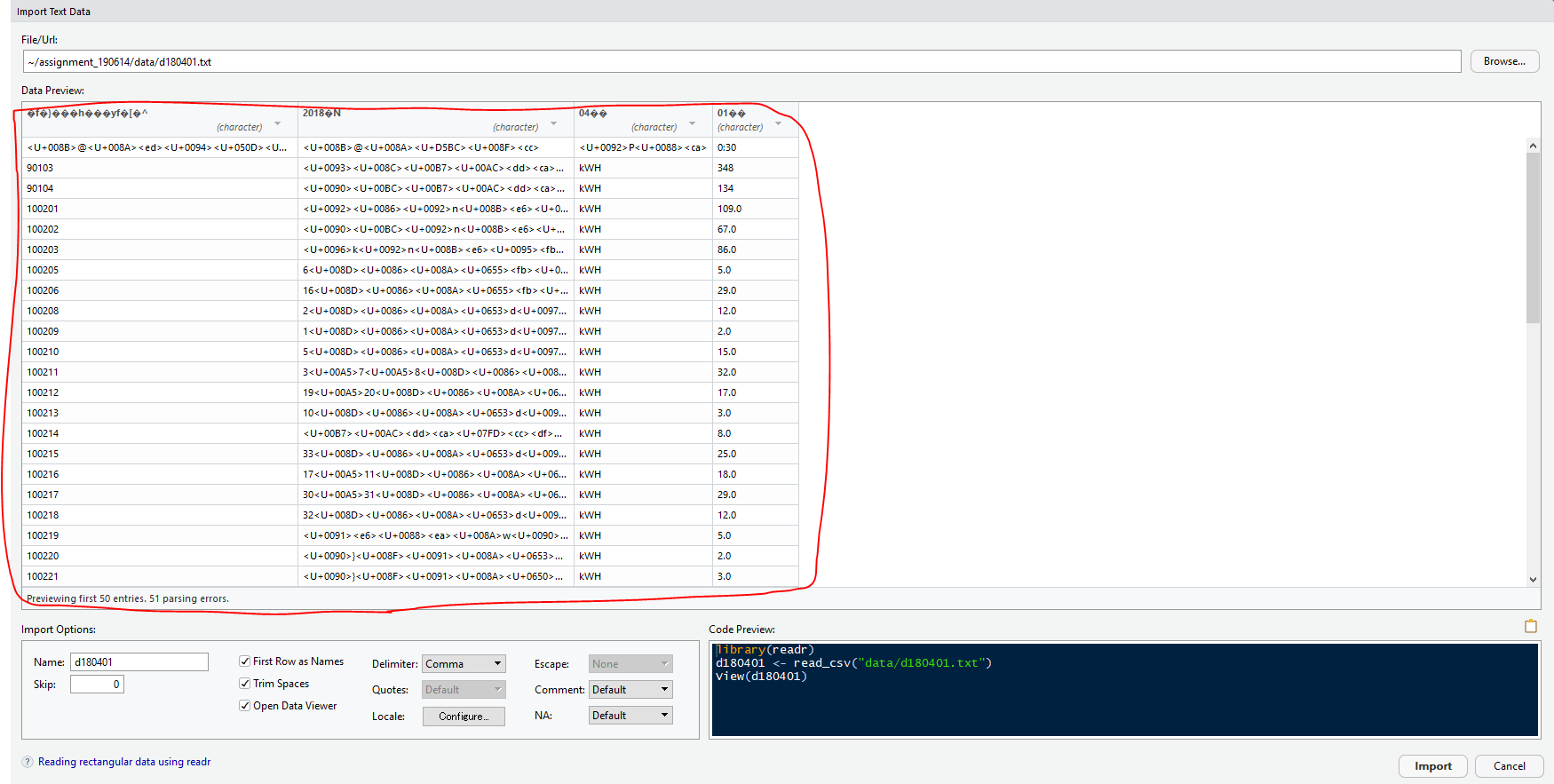
まず右上のWorkspaceパネルにある*import Dataset*ボタンをクリックし、*From Text(readr)*の部分をクリックしてください。Rからreadrパッケージをインストールするよう求められた場合は、Yesを選んでください。



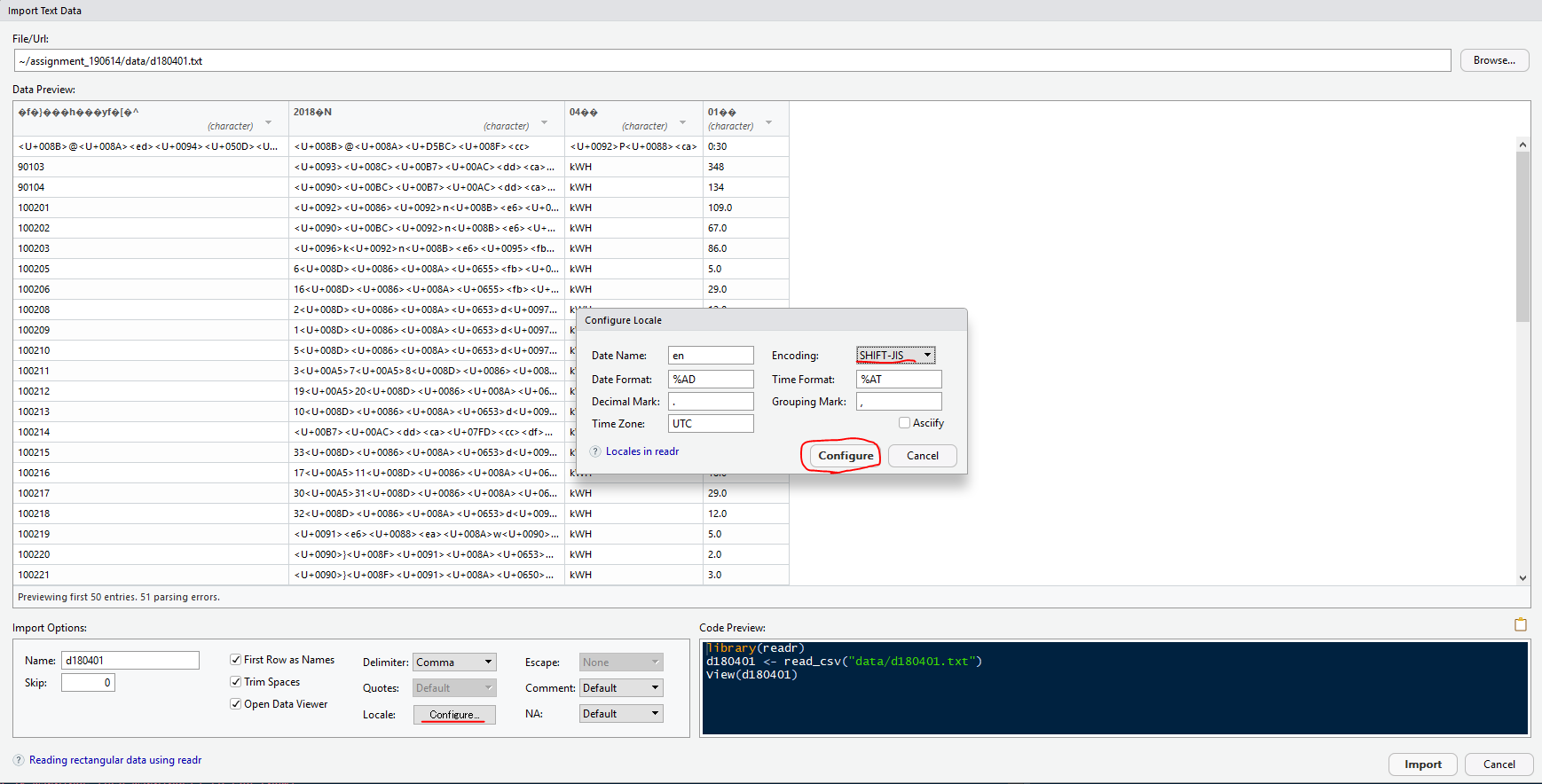
すると、このような画面が立ち上がるので、*Browse*ボタンをクリックし、~/assignment\_190614/dataにあるd180401.txtを開いてみてください。



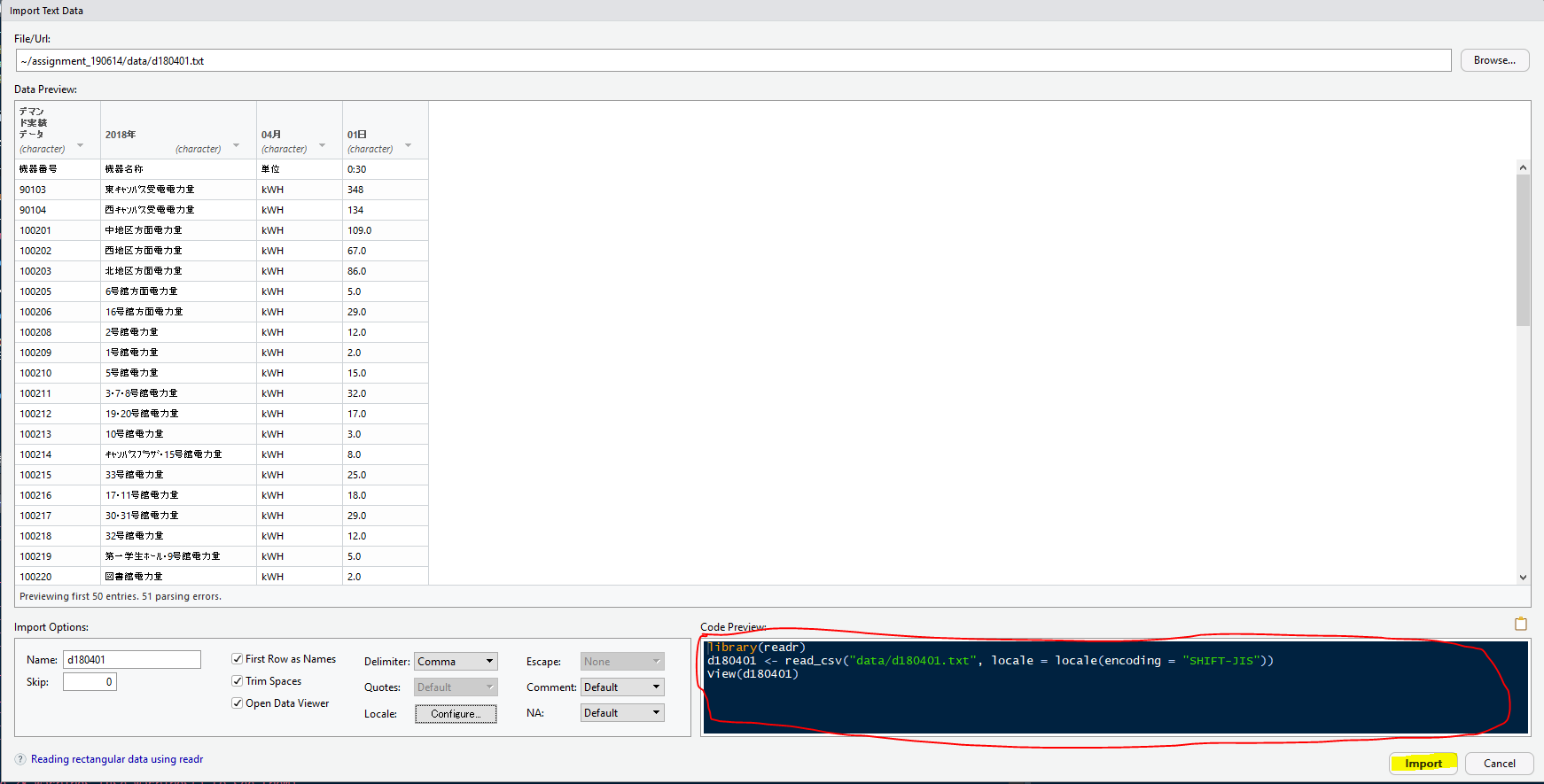
d180401.txtのData Previewを確認すると、列名やその値が正確に読み込めていない（文字化けが生じている）ことが分かります。これは、文字符号化（エンコード）形式が誤っているためです。日本語のPC環境は主に*UTF-8*あるいは*Shift-JIS*が文字符号化形式が利用されているため、最低限この2種類は覚えておいてください。スマートBEMSから得られるデータの文字符号化形式は*Shift-JIS*です。



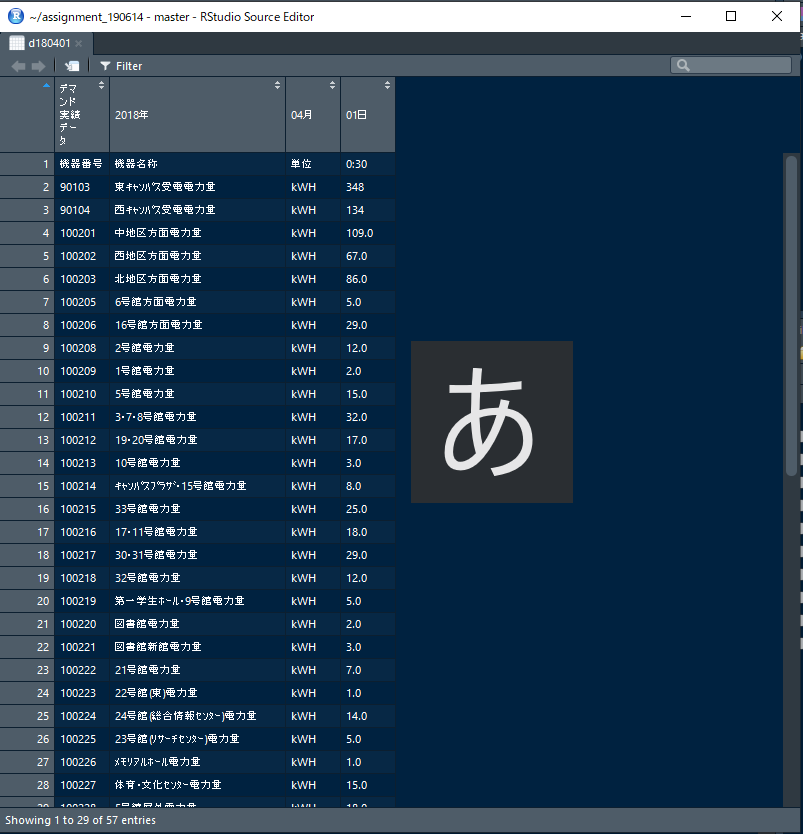
文字符号化形式を変更するには、Locale:の*Configure…*ボタンをクリックし、ポップアップとして表示されるウィンドウのEncodingを*Shift-JIS*に変更し、ポップアップウィンドウ上の*Configure*をクリックしてください。



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。



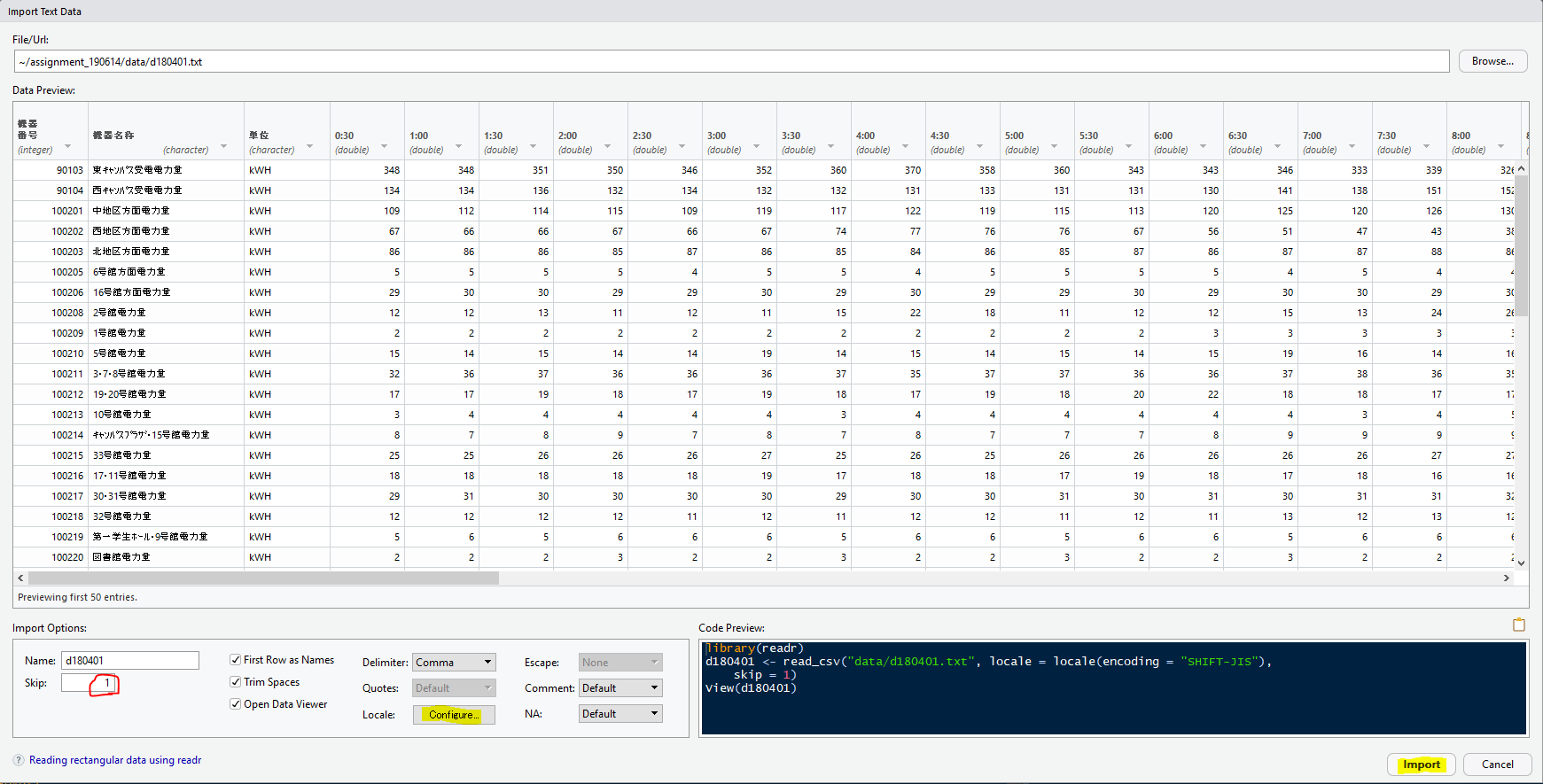
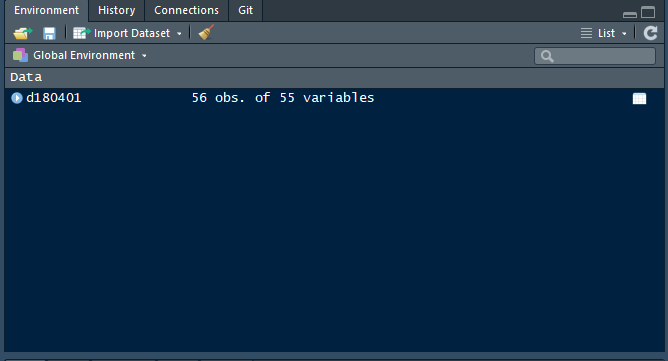
ただ、読み込んだオブジェクトd180401に違和感を覚えないでしょうか？



* 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。

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](http://cse.naro.affrc.go.jp/takezawa/r-tips/r/28.html)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](http://cse.naro.affrc.go.jp/takezawa/r-tips/r/03.html)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.txt"  
## [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$`東ｷｬﾝﾊﾟｽ受電電力量　　　　　　　`  
c2 <- t\_temp$`西ｷｬﾝﾊﾟｽ受電電力量　　　　　　　`  
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$`10号館電力量　　　　　　　　　　`  
c14 <- t\_temp$`ｷｬﾝﾊﾟｽﾌﾟﾗｻﾞ･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$`25号館方面電力量　　　　　　　　`  
c32 <- t\_temp$`ｸﾞﾗｳﾝﾄﾞ方面電力量 　　　　　　　`  
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$`ｸﾞﾗｳﾝﾄﾞ電力量 　　　　　　　　　`  
c42 <- t\_temp$`29号館(ｸﾗﾌﾞ･ｻｰｸﾙﾌﾟﾗｻﾞ)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,c10,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ペすれば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()`

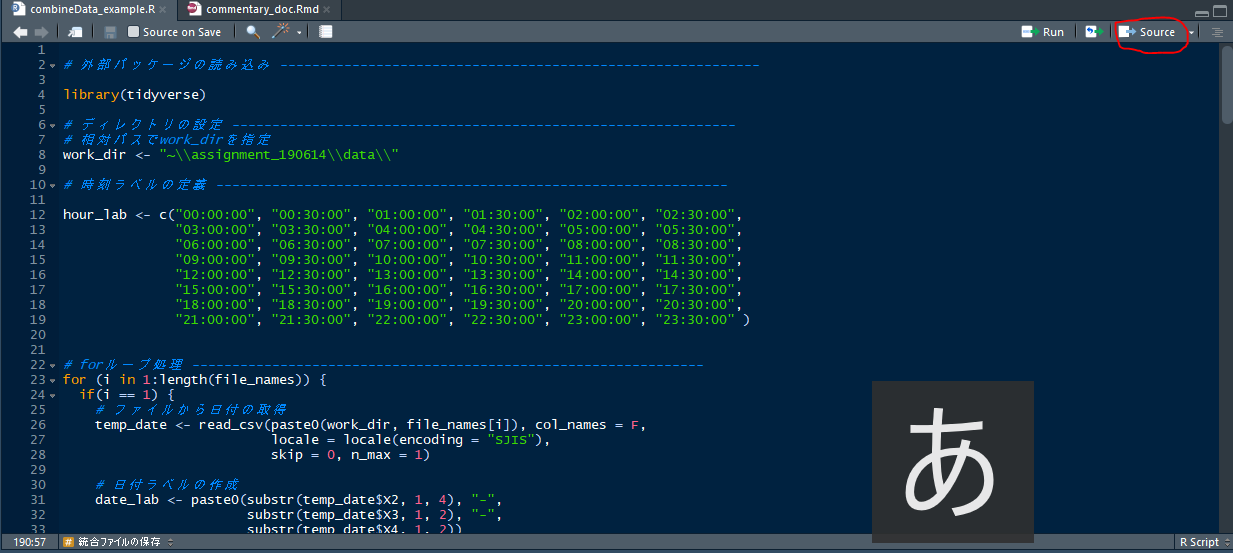
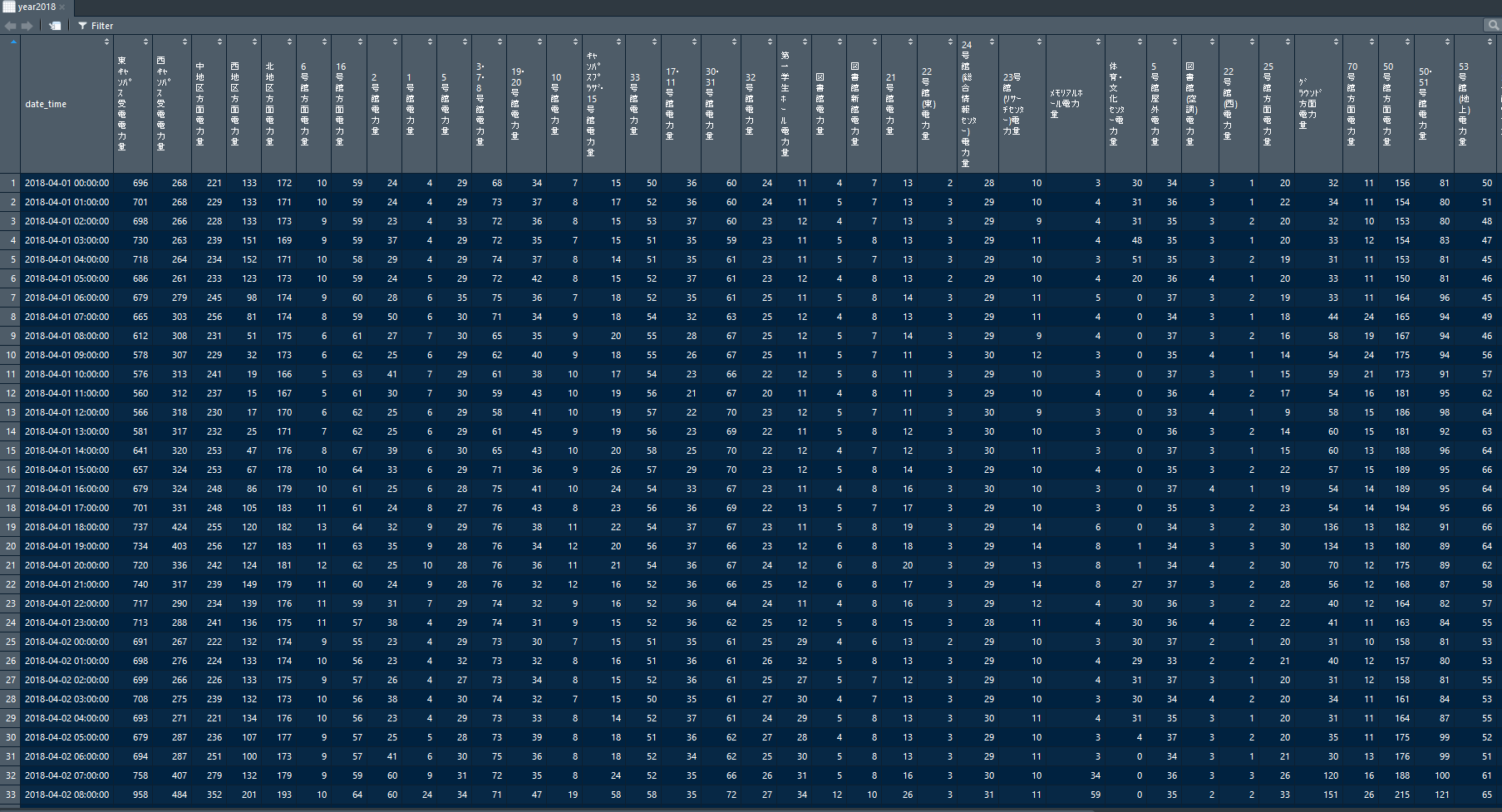
## データの保存

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 editer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。

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！