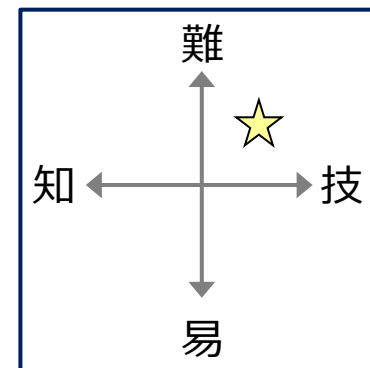


総務省 ICTスキル総合習得教材



[コース4] オープンデータ・ビッグデータ利活用事例



4-3 : プログラミングによるビッグデータの分析 (R)



http://www.soumu.go.jp/ict_skill/pdf/ict_skill_4_3.pdf

	1	2	3	4	5
[コース1] データ収集					
[コース2] データ蓄積					
[コース3] データ分析					
[コース4] データ利活用			●		

本講座の学習内容 [4-3 : プログラミングによるビッグデータの分析 (R)]

【講座概要】

- プログラミング言語のR、統合環境のRStudioの概要と画面構成と基本操作を説明します。
- RStudioを用いたExcelファイルの読み込み方法、Rのパッケージ利用方法を示します。
- Rはパッケージを利用することで、様々な分析や出力が簡単に行えることを示します。
- 一般電気事業者が公開している1時間単位の消費電力ビッグデータをRを用いて分析します。
- 【参考2】においては、Rによるe-Stat APIの利用手順を示します。

【講座構成】

実習

[1] RとRStudioの概要

[2] Rにおける回帰分析・パッケージ利用

[3] Rによるビッグデータの分析

【参考1】 RとRStudioのダウンロード・インストール

【参考2】 Rによるe-Stat APIの利用

【学習のゴール】

- ✓ 統計分析に適したプログラミング言語のRと統合環境のRStudioの概要を把握する。
- ✓ RStudioにおける画面構成、基本操作、パッケージの利用方法を把握し、Rにおける基礎的なプログラミングができるようになる。
- ✓ Rを用いたビッグデータ分析を通して、定量的な効果測定、予測、発見の事例を理解する。

統計分析ソフトウェアRとRStudio

◆この講座では、統計分析用ソフトウェア（プログラミング言語）のRの実習を行います。

- この講座では、様々な分析に利用されているR（アール）を説明し、その活用を実習形式で行います。
 - Rは、Windows、Macintosh、Linuxにインストールできる無料のソフトウェアであるとともにプログラミング言語です。
- Rは、データ分析に特化したプログラミング言語で、データ分析の初心者から専門家まで幅広く人気があります。
 - 様々なソフトウェアの制作に利用されるC言語やJavaといった汎用プログラミング言語と異なり、Rはデータ分析がしやすい設計になっています。
 - 米国電気電子学会が人気のあるプログラミング言語を示した「The Top Programming Languages 2017」において、Rは第6位になっています。
- RStudioは、Rを快適に利用することができる統合開発環境です。
 - 統合開発環境（IDE: Integrated Development Environment）は、一つのソフトウェアの中に入力欄、出力欄、データ欄等が統合されて表示されることで、プログラミング等による開発を行いやすくする環境です。
 - RStudioは、無料で利用できるオープンソース版と優先的なサポートが受けられる商用ライセンスがあります。

統計分析ソフトウェアRのロゴ



© 2016 The R Foundation.

【出所】The R Foundation

<https://www.r-project.org/>

The Top Programming Languages 2017の順位

Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️	100.0
2. C	📱 🖥️ 🖨️	99.7
3. Java	🌐 📱 🖥️	99.4
4. C++	📱 🖥️ 🖨️	97.2
5. C#	🌐 📱 🖥️	88.6
6. R	🖥️	88.1
7. JavaScript	🌐 📱	85.5
8. PHP	🌐	81.4
9. Go	🌐 🖥️	76.1
10. Swift	📱 🖥️	75.3

Rは
第6位
の人気

【出所】米国電気電子学会（IEEE）

<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017>

統合開発環境RStudioのロゴ



RStudio is trademarks of RStudio, Inc

【出所】RStudio, Inc

<https://www.rstudio.com/>

- RおよびRStudioのダウンロード・インストール方法は、本教材の【参考1】に記載しています。
- Rのプログラミングコードを記入したファイルおよび該当する入力コード番号は、各スライドの右上に示しています。

http://www.soumu.go.jp/ict_skill/dc/ict_4_3data_code.zip

Rの起動と基本操作

◆まず、RStudioを使わずにRを直接操作して、プログラミングと出力の関係を確認します。

- RおよびRStudioのインストール後は、右下のようなショートカットアイコンが表示されます。

- 「R i386」は32ビット版のRを指し、「R x64」は64ビット版のRを指します。Windowsの場合は、利用しているWindowsが32ビット版なら「R i386」、64ビット版なら「R x64」を使ってください。利用しているWindowsが、32ビットか64ビットかわからない場合は、どちらでもプログラムが動く「R i386」を使ってください。



- まず、RStudioを使わずに直接Rを操作するために、Rのショートカットアイコンをクリックして起動します。

- Rの基本部分は日本語化済みで、初期画面にはRのライセンスに関する日本語での説明が表示されます。

- Rの直接操作、プログラミング体験として、中央下の**枠内の黒字の部分の入力し、出力を確認**します。

- Rでは「#（番号記号、ナンバーサイン、ハッシュ）」の右側はプログラムとして読み込まないので、「#」の右側には日本語を含めて説明書きやコメントを記入することができます。

Rの初期画面の表示

```

RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> |

```

Rでの基本演算 [入力コード01]

```

#足し算としての「1+2」
1+2
#Rで変数を作る場合は
#「変数名 <- 変数の中身」で入力
# x に10、yに20を入力
x<- 10
y<- 20
#xとyの足し算としてのz
z=x+y
#変数名を入力すると、変数の値を出力
z
#全体を()でくると、計算と同時に出力
(zz=x*y)

```

Rの出力

```

> #足し算としての「1+2」
> 1+2
[1] 3
> #Rで変数を作る場合は
> #「変数名 <- 変数の中身」で入力
> #xに10、yに20を入力
> x<- 10
> y<- 20
> #xとyの足し算としてのz
> z=x+y
> #変数名を入力すると、変数の値を出力
> z
[1] 30
> #全体を()でくると、計算と同時に出力
> (zz=x*y)
[1] 200

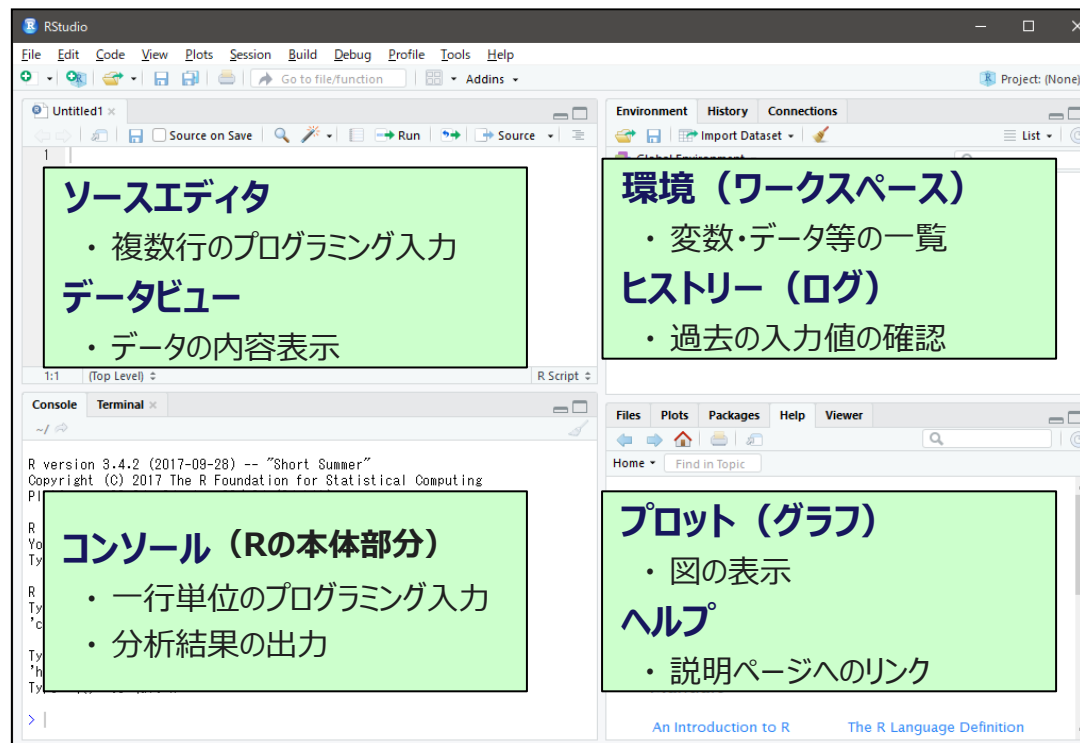
```

RStudioの画面構成

◆ RStudioは分割した画面構成によって、Rのプログラミングを効率的に行えます。

- RStudioのショートカットアイコンをクリックすると、**分割された画面構成のRStudio**が起動します。
 - RStudioには、公式の日本語版はありませんが、初歩的な英単語の知識で概ね読むことができます。また、ウェブ上の無料翻訳サービスを活用すれば、英語が苦手でもRStudioの利用に支障はありません。
 - もし画面の左側が縦に分割されていない場合は、画面上側のメニューの左端にある[File]→[New File] →[R Script] を選択してください。
- RStudio内では分割された各パネルで**入力欄、出力欄、データ表示欄、グラフ表示欄**と機能が分けられています。
 - RStudioでは分割された各パネルにタブ（つまみボタン）が付いており、パネル内の表示内容や表示対象を変えることができます。
 - RStudioの画面構成は、メニューの[Tools]→[Global Options] →[Panel Layout]から、利用者の好みに合うようにカスタマイズできます。

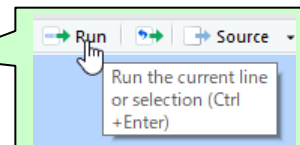
初期設定におけるRStudioの画面構成（主なタブの内容）



RStudioへの入力と画面出力

◆RStudioを使うと、変数データ一覧やグラフを確認しながら、プログラミングができます。

- 右下の画像では、ベクトル形式のデータを操作、線付きの散布図（グラフ）の描画を行っています。
 - RStudioでは、4分割の左上にあるソースエディタから実行したいコードの範囲を選択後、「Run」のボタンをクリックしてください。
 - ベクトルは、数値を横（行）または列（縦）に並べたものを指し、数値を束ねたもののイメージです。
 - Excel同様、括弧内に対象を指定して定められた処理をするものを関数といいます。例えば、**mean**関数は平均値を導出します。



基本統計量の導出等 [入力コード02]

RStudioの4分割画面の表示

2種類のベクトルの記入

```
v1<- c(1, 2, 3, 2, 1)
```

```
v2<- c(10, 20, 30, 40, 50)
```

#ベクトル同士の足し算（表示付）

```
(plus_v1v2=v1+v2)
```

#2つのベクトルを横に並べて行列作成（表示付）

```
(set_v1v2=cbind(v1, v2))
```

統計関数の利用

#平均値mean

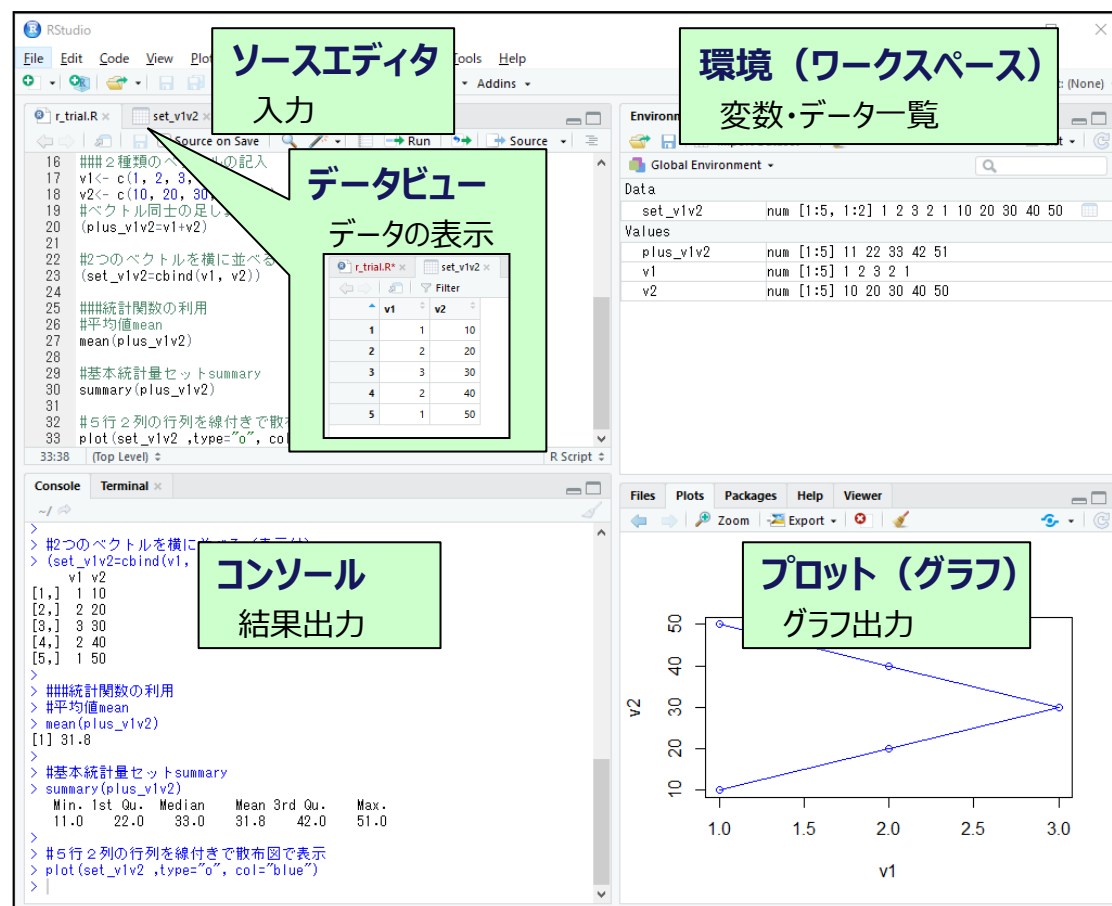
```
mean(plus_v1v2)
```

#基本統計量セットsummary

```
summary(plus_v1v2)
```

#「set_v1v2」を線付きで散布図で青で表示

```
plot(set_v1v2 ,type="o", col="blue")
```

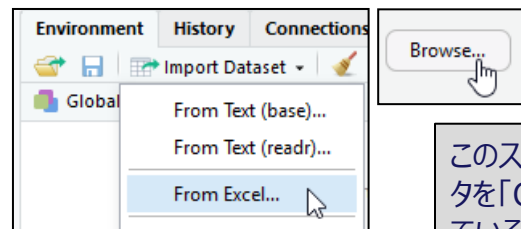


RStudioにおけるExcelファイルの読み込み

◆ RStudioでは、Excelデータをマウス操作で簡単に読み込むことができます。

- RStudioの標準設定における右上のパネルの[Import Dataset]から外部のデータを読み込みます。
- 実習用Excelファイルの取り込む場合は[From Excel]→[Browse]とクリックし、「tokyo_drink.xlsx」の選択後、プレビューでデータの内容を確認してから [Import]をクリックしてください。
 - Excelファイル内の分析用データは1行目に変数名、2行目以降に一行ずつ個別の標本のデータが入っている形式にしておきます。
 - ファイル名が日本語だと取り込む際にエラーになることがあるので、半角英数字のファイル名にしておくと、読み込み時のエラーの心配がありません。一方、Excelファイルの中身の各セルに入っているデータは、文字データの列であれば日本語が含まれていても問題ありません。
 - データがプレビューに表示されている状態では、[Code Preview] にデータと読み込みに対応するコードが表示されます。このコードをコピーして、ソースエディタに貼りつけることで、次回以降の同じデータ読み込みをする際にはマウスによる操作は不要でコード内で行えます。

Excelファイルの指定と[Browse]



このスライドでは、実習用データを「C:/data/」下に設置している設定で説明します。

プレビューによるデータ内容の確認

File/Url: C:/data/tokyo_drink.xlsx			
Data Preview:			
日付 (character)	曜日 (character)	気温 (double)	湿度 (double)
2016/1/1	金(休)	7.5	45
2016/1/2	土(休)	7.3	51

講座3-4の「Excelによる回帰分析」で利用したデータと内容は同じです。

コードとしての表示と[Import]

```
Code Preview:
library(readxl)
tokyo_drink <- read_excel("C:/data/tokyo_drink.xlsx")
View(tokyo_drink)
```

コードに記入してExcelを読み込む場合は、このコードをソースエディタに貼りつけます。(Viewで始まる行は不要です。)

Import Cancel



RStudio内に取り込んだExcelデータの表示

tokyo_drink						
	日付	曜日 (休日)	気温	湿度	土日祝 ダミー	飲料 販売量
1	2016/1/1	金(休)	7.5	45	1	296
2	2016/1/2	土(休)	7.3	51	1	414
3	2016/1/3	日(休)	9.3	61	1	343
4	2016/1/4	月	9.2	60	0	514
5	2016/1/5	火	10.9	51	0	429
6	2016/1/6	水	8.9	69	0	478
7	2016/1/7	木	8.7	49	0	384

Environment History Connections
Global Environment
Data
tokyo_drink 366 obs
日付: chr "2016/1/1" "2016/1/2" "2016/1/3" "2016/1/4" "2016/1/5" "2016/1/6" "2016/1/7"
曜日(休日): chr "金(休)" "土(休)" "日(休)" "月" "火" "水" "木"
気温: num 7.5 7.3 9.3 9.2 10.9 8.9 8.7
湿度: num 45 51 61 60 51 69 49
土日祝ダミー: num 1 1 1 0 0 0 0
飲料販売量: num 296 414 343 514 429 478 384

Rにおける回帰分析

◆Rでは読み込んだデータに対して、1行のコードで回帰分析が実行できます。

- RStudioから読み込んだExcelファイルは、**データフレーム**と呼ばれるデータの形式となり、データフレーム内の各列は「**データフレーム名\$列名（変数名）**」で指定することができます。
 - ・ RStudioの標準的なExcelファイルの読み込み設定では、Excelファイル上のデータの1行目が、Rにおける列名（変数名）となります。
- Rにおける回帰分析は、「**lm(被説明変数 ~ 説明変数1 + 説明変数2 + ...)**」という1行のコードで実行できます。
 -  プログラミングをしやすいように短いデータフレーム名「td」へとコピーするために「**td<-tokyo_drink**」と入力します。
 -  回帰分析の結果を「lm_result」という名前のデータ(リスト形式)として格納するために、「**lm_result<-lm(td\$飲料販売量 ~ td\$気温 + td\$湿度 + td\$土日祝ダミー)**」と入力します。
- 回帰分析の結果データに対して、**summary**関数を使うと、回帰分析の結果が表示されます。
 - ・ **summary**関数は、対象にデータ自体を指定すると基本統計量を表示しますが、回帰分析の結果を指定すると、分析結果を表示します。



講座3-4と同一の回帰分析 [入力コード03後半]

#データフレーム名を短くして、新たにデータフレームを作成

td<-tokyo_drink

#td内の飲料販売量を被説明変数、気温と湿度と土日祝ダミーを説明変数として回帰分析

lm_result<-lm(td\$飲料販売量~td\$気温+td\$湿度+td\$土日祝ダミー)

#決定係数などの結果確認

summary(lm_result)

Rの統計的検定においては
両側5%で有意なら「*」、
両側1%で有意なら「**」、
両側0.1%で有意なら「***」
が表示されます。

環境（ワークスペース）における表示

Data	
data_ols	50 obs. of 4 variables
lm_result	List of 12

講座3-4におけるExcelの分析ツール
での回帰分析と同じ結果です。

Summaryによって示される回帰分析の説明変数の出力

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	374.9633	14.5686	25.738	<2e-16 ***
td\$気温	5.8803	0.4914	11.965	<2e-16 ***
td\$湿度	0.4903	0.2318	2.115	0.0351 *
td\$土日祝ダミー	-86.7386	6.9676	-12.449	<2e-16 ***

Rにおけるパッケージの利用

◆Rでは、ウェブ上に公開されているプログラム（パッケージ）を利用することができます。

- 公開されているRのプログラム（パッケージ）を利用すると、簡単な入力で様々な出力・分析を行うことができます。
 - Rにおいて、有用なプログラミングコードを配布用にとりまとめて公開しているものを「パッケージ」といいます。
 - インターネット上のCRANに保存されているパッケージを初めて使う場合は、コードに「`install.packages("パッケージ名")`」と入力し、PC内にパッケージをダウンロード・インストールしてください。（一度、PCにインストールすれば、2回目以降のコードへの記載は不要です。）
 - CRANは原則として英語での表記ですが、CRAN内を検索してパッケージを探すこともできます。（<https://cran.ism.ac.jp/search.html>）
- PCにインストール済みのパッケージは、コードに「`library(パッケージ名)`」と入力した後に使うことができます。
 - `library`は、「図書室」「書庫」「蔵書」を意味し、Rでは`library`関数によって、インストール済みのパッケージを読み込みます。
- 複数の回帰分析の結果を並べて表示して、比較したい場合には、`memisc`パッケージが便利です。
 - `memisc`パッケージに含まれる`mtable`関数に対して、回帰分析の結果データをコンマで区切って指定してください。



パッケージを利用した回帰分析の比較表示 [入力コード04]

```
#回帰式から気温と湿度の各変数を外して、それぞれ2変数で回帰して結果をlm_res2、lm_res3に格納
lm_res2<-lm(td$飲料販売量~td$気温+td$土日祝ダミー)
lm_res3<-lm(td$飲料販売量~td$湿度+td$土日祝ダミー)

#パッケージ「memisc」のインストールと利用宣言（installで始まる行は1度目のみ[要インターネット接続]）
#install.packages("memisc")
library(memisc)

#パッケージmemisc内のmtable関数を利用
#3つの回帰分析の結果を並べて表示
mtable(lm_result, lm_res2, lm_res3)
```

memiscパッケージによる回帰分析の比較出力

◆ Rはパッケージを利用することで様々な出力、高度な分析を簡単に実行できます。

- memiscパッケージに含まれるmtable関数では、回帰分析の結果を並べて分かりやすく表示できます。

memiscパッケージ (mtable関数) による回帰分析の比較出力

回帰分析の結果表示において、(Intercept)は、切片の高さを表し、説明変数の値が全て0の場合における被説明変数の予測値に対応します。

説明変数に関する出力において、括弧のない値は「推定係数」を表し、括弧に入った値は「推定係数の標準誤差」を表しています。

「R-squared」は決定係数を意味し、講座3-4においても示したように0以上1以下の値をとる回帰分析の当てはまり度合いの指標です。
「adj. R-squared」は自由度調整済み決定係数です。

	lm_result	lm_res2	lm_res3
(Intercept)	374.963*** (14.569)	400.563*** (8.151)	376.813*** (17.185)
td\$気温	5.880*** (0.491)	6.394*** (0.429)	
td\$湿度	0.490* (0.232)		1.860*** (0.238)
td\$土日祝ダミー	-86.739*** (6.968)	-87.746*** (6.984)	-84.283*** (8.216)
R-squared	0.518	0.512	0.328
adj. R-squared	0.514	0.510	0.324
sigma	62.689	62.989	73.953
F	129.849	190.710	88.521
p	0.000	0.000	0.000
Log-likelihood	-2031.898	-2034.147	-2092.882
Deviance	1422640.504	1440228.237	1985284.010
AIC	4073.796	4076.293	4193.765
BIC	4093.310	4091.904	4209.375
N	366	366	366

決定係数を横並びで比較することで「回帰分析の当てはまり」に重要な役割を持つ説明変数を把握しやすくなります。

Rによる消費電力ビッグデータの分析

◆このパートではRを用いて、ビッグデータ（1時間単位の消費電力）の分析事例を示します。

● Rをはじめとするプログラミングでは、大容量のビッグデータのデータ処理・分析を行うことができます。

- ・プログラミングによる分析作業は、マウス等による対象データ範囲の指定が不要であるため、一般に標本数やデータサイズに依存しません。
- ・64ビット版のRでは、メインメモリの容量を上限としてデータを格納することができ、GB（ギガバイト）単位のデータ処理が可能です。

● このパートでは、9つの電力会社管内における**1時間単位の消費電力データ・気象データ**を分析に利用します。

- ・沖縄電力を除く9つの一般電気事業者が運営する電力系統は、電力を融通し合う連系線によって近隣の電力系統と結ばれており、本分析は効率的な電力の融通の検討にも利用できます。
- ・分析に利用する消費電力データおよび気象に関するデータは、誰でも下記のURLからダウンロードすることができます。

分析実習用データの提供主体、データにつながるURL

提供主体	分析用データにつながるURL	提供主体	分析用データにつながるURL
北海道電力	http://denkiyoho.hepco.co.jp/area_download.html	関西電力	http://www.kepco.co.jp/energy_supply/supply/denkiyoho/
東北電力	http://setsuden.tohoku-epco.co.jp/download.html	中国電力	http://www.energia.co.jp/jukyuu/
東京電力	http://www.tepco.co.jp/forecast/html/download-j.html	四国電力	http://www.yonden.co.jp/denkiyoho/download.html
中部電力	http://denki-yoho.chuden.jp/	九州電力	https://www.kyuden.co.jp/power_usages/pc.html
北陸電力	http://www.rikuden.co.jp/denki-yoho/#download	気象庁	http://www.data.jma.go.jp/obd/stats/etrn/index.php

● 実習用データの対象期間は**2016年4月1日～2017年12月31日**であり、**15,360時間分**のデータです。

- ・電力自由化の関係で2016年4月より一般電気事業者が提供するデータが変化したため、2016年4月を分析用データの期初としています。
- ・このパートで利用する消費電力・気象データは、大規模標本のビックデータであるとともに改変を伴う利用や再配布ができるオープンデータでもあります。

● 分析作業を効率的に進めるために、下記のような**分析目的を設定**して、データ分析に取り組みます。

- ◆ 気象条件が消費電力に与える影響を分析し、気象条件の設定が与えられた場合に消費電力を予測したい。
- ◆ 寒すぎたり、暑すぎたりすると消費電力が上がるが見込まれるが、消費電力を最小化する気温を知りたい。

消費電力ビッグデータのRへの取り込み

◆ 実習用のExcelファイルを取り込んで、気象等が消費電力に与える影響を分析します。

- 日時別の気象・消費電力データが格納されたExcelファイル「`elec_weather.xlsx`」をRへ取り込みます。
 - 様々な形態のデータを収集、整理統合することも、データ分析における重要なプロセスですが、ここでは省略して実習用のExcelファイルを取り込みます。
 - Excelファイルには9つの電力会社管内に関する消費電力（1時間単位：万kW）、気温（1時間単位：℃）、降水量（1日合計値：mm）のデータが並んでいます。なお、気象データは9つの一般電気事業者の本社所在地を対象地点としています。
 - 1時間単位の消費電力データは、その時点を期初とした後1時間のデータを示しますが、気象庁が公開する1時間単位の気象データは、その時点を期末とした前1時間のデータを示します。このため、Excelファイルでは消費電力データの表記に合わせて、公開された気象データを1時間前にずらしています。

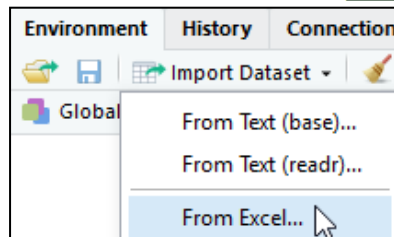
実習用データ「elec_weather.xlsx」の内容（東京電力に関する列）

	A	B	C	D	E	G	H	K	T	AC
1	年月日時	年月	月	年月日	曜日	平日or土日祝	時	東京電力 (1時間単位)	東京_気温 (1時間単位)	東京_降水量 (1日合計値)
2	2016/4/1 0時	201604	4	2016/4/1	金	平日	0	2555	13.8	2.5
3	2016/4/1 1時	201604	4	2016/4/1	金	平日	1	2433	13.0	2.5
4	2016/4/1 2時	201604	4	2016/4/1	金	平日	2	2393	12.2	2.5
5	2016/4/1 3時	201604	4	2016/4/1	金	平日	3	2375	11.2	2.5

- RStudioの[Import Dataset]ボタンまたは下記のコードによって、実習用データのExcelファイルを取り込みます。



実習用データ「elec_weather.xlsx」のRへの取り込み [入力コード05]



1時間単位の消費電力、気象データのExcelファイルの取り込み

```
library(readxl)
```

```
elec_weather <- read_excel("C:/data/elec_weather.xlsx")
```

このスライドでは、実習用データを「C:/data/」下に設置している設定で説明します。

分析用データの抽出と変数名の改訂

◆ 分析に用いる変数を抽出し、プログラミングしやすいように変数名を英字に改訂します。

- Excelから取り込んだ全てのデータは、データフレーム「`elec_weather`」に入っていますが、日時のデータおよび回帰分析の対象とする地域のデータをデータフレーム「`elwe`」へと抽出します。
 - Rでは『**新たなデータフレーム名** <- **抽出元のデータフレーム名** [, **抽出したい変数名のベクトル**]』とコードを書くことによって、変数を抽出して新たなデータフレームを作成することができます。
- プログラミングをしやすいように**日本語の変数名を半角英数字の変数名へと変換**します。
 - Rでは日本語の変数名を取り扱えますが、プログラミングにおいては半角英数字の変数名が一般的であり、短い変数名の方が見やすいコードが書けます。
 - Rでは、『**name(データフレーム名)** <- **c("変数名1", "変数名2", ...)**』とコードを書くことによって、データフレームの左から新たな変数名を指定できます。



分析対象地域の電力に関するデータの抽出と変数名の改訂（東京電力のケース） [入力コード06]

日時データおよび東京電力データに関するデータの変数名のベクトルで指定

```
lm_name <- c("年月日", "月", "平日or土日祝", "時", "東京電力（1時間単位）", "東京_気温（1時間単位）", "東京_降水量（1日合計値）")
```

東京電力データに関する変数の列を抽出して、新たなデータフレームelweを作成

```
elwe <- elec_weather[, lm_name]
```

プログラミングしやすいように変数名（列名）を英数字へ

```
names(elwe) <- c("ymd", "month", "hei_dns", "hour", "elec", "atemp", "wdrop")
```

- 上記のコード例では、東京電力管内の消費電力に着目して、東京電力に関する変数を抽出しています。
 - プログラミングコードのファイルは地域別に9種を用意していますが、スライドでは東京電力管内に対応する「code_2_elec3tokyo.R」を用いて説明します。
 - 他の地域に関する分析を参照する場合は、コード内の（東京電力, 東京_）の部分（北海道電力, 札幌_）（東北電力, 仙台_）（中部電力, 名古屋_）（北陸電力, 富山_）（関西電力, 大阪_）（中国電力, 広島_）（四国電力, 高松_）（九州電力, 福岡_）へと変更するか、各地域に該当するコードが記入されたファイルを参照してください。

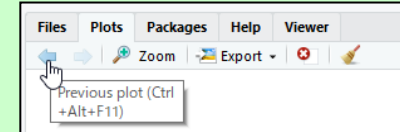
散布図による変数間の関係の確認

◆ 散布図によって、気象に関するデータと消費電力データの関係を確認します。

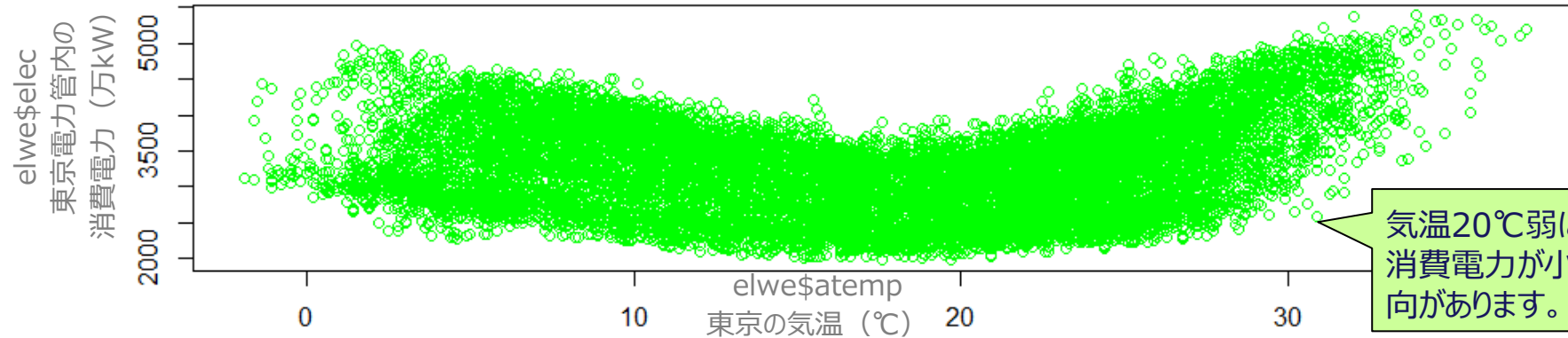
「気温と消費電力」「降水量と消費電力」の散布図の作成 [入力コード07]

```
#「気温と消費電力(緑色)」「降水量と消費電力(青色)」のそれぞれの散布図の作成
plot(elwe$atemp, elwe$elec, col="green")
plot(elwe$wdrop, elwe$elec, col="blue")
```

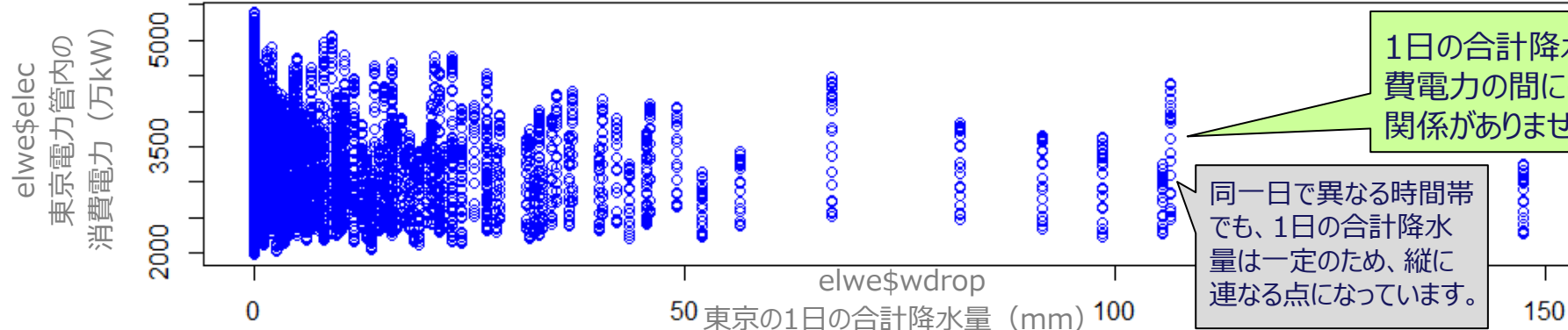
RStudioでは、複数の図を出力しても、矢印ボタンで表示を移すことができます。



東京の気温 (°C) と東京電力管内における消費電力 (万kW)



東京の1日の合計降水量 (mm) と東京電力管内における消費電力 (万kW)



□ 降水があると、気温が下がる傾向にあるため、散布図や単回帰分析では、各変数の正確な効果を把握できません。

累乗項・ダミー変数を説明変数とする重回帰分析

◆ 散布図を踏まえて、説明変数を設定して重回帰分析を行います。

- 気温と消費電力の非線形の関係を表すため、気温を2乗および3乗した値を説明変数に加えます。
 - 同一の説明変数を2乗すれば2乗項、3乗すれば3乗項となり、総じて累乗項といいます。
- 連続値としての降水量は、消費電力との明瞭な関係が見られないため、ダミー変数として「降水の有無」を設定します。
 - 該当する場合を1、該当しない場合を0と表し、有無や該当・非該当の状態を0と1で表した変数をダミー変数といいます。
 - Rにおけるifelse関数は、『ifelse([条件], [条件に合う場合の出力], [条件に合わない場合の出力])』と記述して、ダミー変数の作成に利用できます。
- 回帰分析の結果の推定係数を出力し、CSVファイルに保存します。
 - Rにおけるcoef関数は括弧内に回帰分析の結果データを指定することで、回帰分析の推定係数を表示できます。
 - Rにおけるwrite.table関数は、『write.table([R内の保存対象], "[PC内の出力先]", sep="[区切り文字]")』と記述して、ファイル保存ができます。



気候データを用いた回帰分析の実行と推定係数の表示と保存 [入力コード08]

```
#地域の気温の2乗項、3乗項を作成する
```

```
elwe$atemp2<- elwe$atemp^2
```

```
elwe$atemp3<- elwe$atemp^3
```

```
#1日の降水有無のダミー変数を作成する
```

```
elwe$wadr_dummy<- ifelse(elwe$wdrop>0, 1, 0)
```

```
#地域の電力利用を気温（3乗項まで）と降水有無のダミー変数で回帰分析を行い、lm_re1へ結果データを保存
```

```
lm_re1<-lm(elwe$elec~elwe$atemp+elwe$atemp2+elwe$atemp3+elwe$wadr_dummy)
```

```
#回帰分析の結果データより推定係数を表示
```

```
coef(lm_re1)
```

```
#回帰分析の推定係数をCSVファイルに保存
```

```
write.table(coef(lm_re1), "C:/data/lm_re1_coef.csv", sep=",")
```

推定係数の一覧をコンマ区切りで「lm_re1_coef.csv」へ保存します。

「気温」と「降水の有無」による重回帰分析の結果表示

◆ 重回帰分析の分析結果の推定係数を確認します。

- 消費電力を「3乗項までの気温の累乗項」「降水の有無のダミー変数」で回帰分析を行った結果の推定係数として、下記のような推定値が得られます。

```
> coef(lm_re1)
(Intercept)      elwe$atemp      elwe$atemp2      elwe$atemp3      elwe$wadr_dummy
3453.6410544      35.4280133      -8.5273962       0.2650495       68.5061220
```

- カンマで区切られたテキストファイルに当たるCSVファイルにおいても、推定係数を出力することができます。

- CSVファイルは、Windowsのメモ帳のようなテキストエディターでも、Excelでも開くことができます。

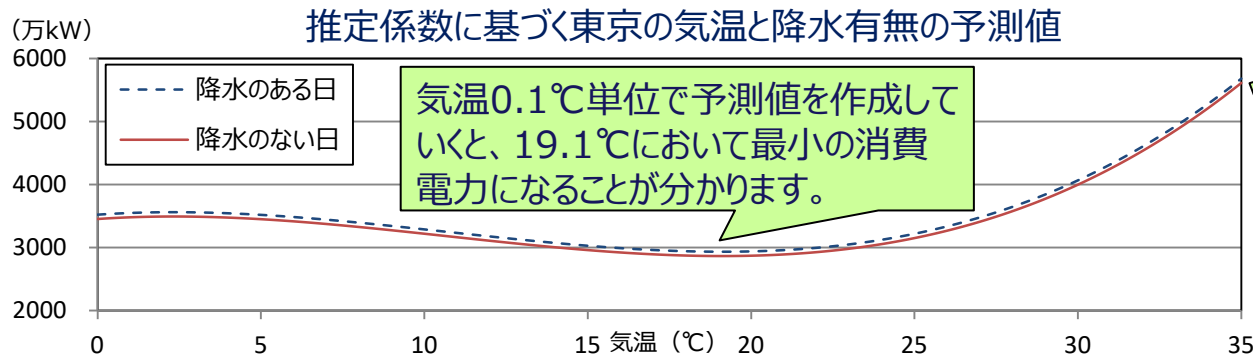


```
"x"↓
"(Intercept)",3453.6410544404↓
"elwe$atemp",35.4280132539264↓
"elwe$atemp2",-8.52739622209195↓
"elwe$atemp3",0.265049482073631↓
"elwe$wadr_dummy",68.5061220158637↓
```

	A	B
1	x	
2	(Intercept)	3453.641
3	elwe\$atemp	35.42801
4	elwe\$atemp2	-8.5274
5	elwe\$atemp3	0.265049
6	elwe\$wadr_dummy	68.50612

- 推定係数に基づいて、各気温の当てはめ値を導出し最小の消費電力をもたらす気温を導出することができます。

- 『Intercept+atemp・気温+atemp2・気温²+atemp3・気温³+wadr_dummy・降水有無の設定』で気温ごと・降水別の予測値が得られます。



「降水のある日」は「降水のない日」に比べて68.5万kW消費電力が大きくなる予測値がダミー変数の推定係数から示されます。

例示した回帰分析の式の形（ダミー変数の入れ方）では、1日の降水の有無によって予測値が上下に並行移動する形になっています。

月別・時間帯別の消費電力の箱ひげ図

◆ 月別・時間帯別の消費電力を箱ひげ図によって確認します。

- 月別・時間帯別の消費電力を見るために、データの種別を変換し、月別・時間帯別の箱ひげ図を描きます。



日付に関するデータの変換と箱ひげ図の作成 [入力コード09]

休日なら1、平日なら0のダミー変数を作成する

```
elwe$off_dummy <- ifelse(elwe$hei_dns=="土日祝", 1, 0)
```

Ifelse関数内でとイコールを2つ連ねることで、文字列が等しい条件を表します。

plot関数で箱ひげ図を描くためにも、

後にダミー変数を作成するmakedummiesパッケージを利用するためにも

月と時間帯のデータの型を因子型 (factor) へ変更

```
elwe$month <- as.factor(elwe$month)
```

```
elwe$hour <- as.factor(elwe$hour)
```

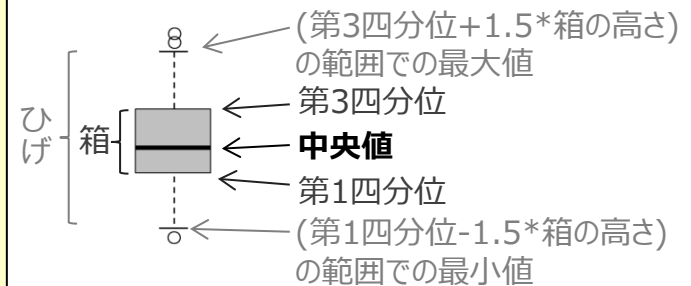
「因子型」とは、カテゴリーを表すデータの型を指しています。

plot関数は因子型データを一つ目、連続データ二つ目に入れると箱ひげ図を表示

```
plot(elwe$month, elwe$elec, col="green")
```

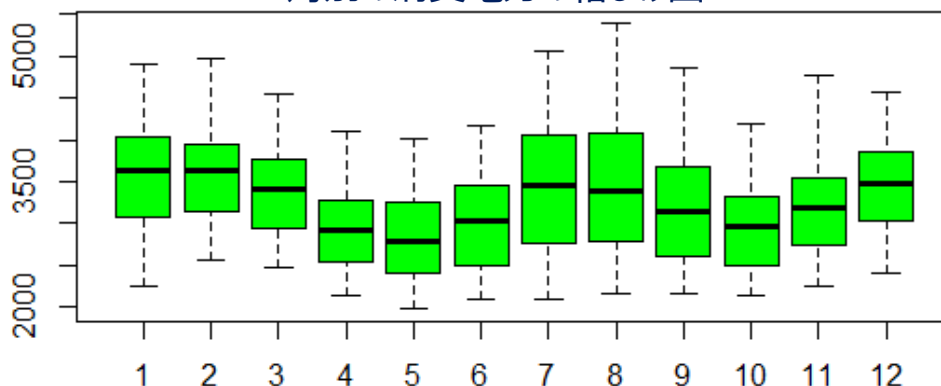
```
plot(elwe$hour, elwe$elec, col="cyan")
```

Rのplotにおける箱ひげ図の表記

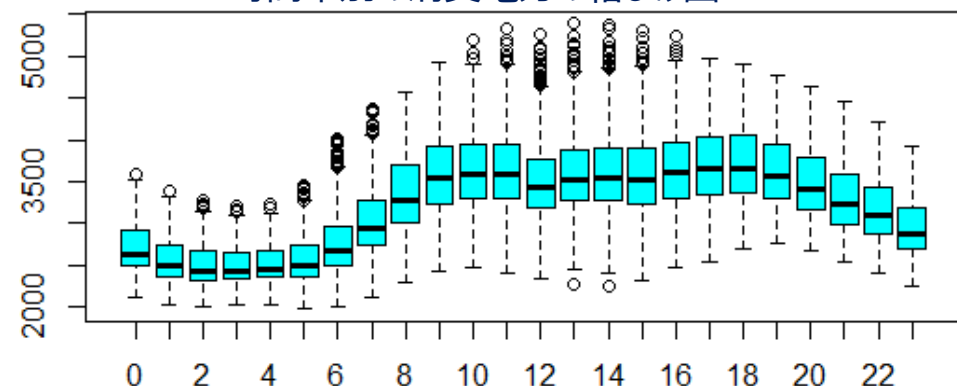


※ Rの箱ひげ図では「ひげ」の範囲外となる値を「外れ値」として、○で示します。

月別の消費電力の箱ひげ図



時間帯別の消費電力の箱ひげ図



- 夏・冬の消費電力が春・秋に比べて大きい傾向、日中の消費電力が夜間に比べて大きい傾向が示されています。

月別・時間帯別のダミー変数の作成

◆ 回帰分析において説明変数として用いる 月別・時間帯別のダミー変数を作成します。

- **makedummies**パッケージを利用して、月別・時間帯別のダミー変数を作ります。
 - **makedummies**パッケージの利用において、初期値でもある『**basal_level=FALSE**』にすることで、最も小さい値（例：1月、0時～）のダミー変数を作成しません。一方、『**basal_level=TRUE**』にすると、最も小さい値のダミー変数を含めて作成します。
 - 最も小さい値のダミー変数を含めて回帰分析を行うと、他の説明変数の定数倍と和が特定の説明変数と等しくなる「完全な多重共線性」という問題が生じ、回帰分析ができなくなります。Rの**lm**関数では「完全な多重共線性」があっても、当該説明変数を自動で外して回帰分析を行います。ここではあらかじめ、最も小さい値のダミー変数を作成しません。



makedummiesパッケージによるダミー変数の作成 [入力コード10]

```
#install.packages("makedummies")
library(makedummies)

#makedummiesを使って月ダミーと時間帯ダミーを作成する
#「basal_level = FALSE」にすると、一番小さい値の1月および0時のダミー変数を作成しない。
md<-makedummies(elwe, basal_level = FALSE, col = "month")
hd<-makedummies(elwe, basal_level = FALSE, col = "hour")

#月ダミーと時間帯ダミーの統合
mdhd<-data.frame(md, hd)
```

- 環境（ワークスペース）に表示されたデータをクリックすることで、作成したダミー変数の内容を確認できます。
 - 『**時間・月ダミー<-data.frame(月ダミー, 時間ダミー)**』と記述し、2種類のダミーデータを横に並べた時間・月ダミーのデータフレームを作成しています。



データをクリック

▶ hd
▶ md
▶ mdhd

月別のダミー変数(md)

	month_2	month_3	month_4	month_5
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	0	1	0

時間帯別のダミー変数 (hd)

	hour_1	hour_2	hour_3	hour_4
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0

日時に関するダミー変数を含めた回帰分析の実行

◆ 休日・月別・時間帯別のダミー変数を含めて消費電力の回帰分析を行います。

- 日時に関するダミー変数を用いて、3種類の回帰分析を行います。
 - `lm`関数において、説明変数に『+ .』を記入すれば、『, data=』の後に記入したデータセットを全て説明変数に含めます。ここでは、`makedummies`パッケージを用いて作成した月ダミーセットの`md`、時間ダミーセットの`hd`、月・時間ダミーセットの`mdhd`をそれぞれ説明変数に加えます。
- これまでの回帰分析の結果を`memisc`パッケージの`mtable`関数を使って比較できる形で出力します。
 - 下記のコードにおいては、R内の出力とCSVファイルに保存する方法をそれぞれ示しています。
 - `mtable`の初期設定では推定係数にP値に基づく「*」がついており、CSV等で数値としての取り扱いの支障になることもあります。「*」を除いた出力にするには、`mtable`関数内で『,signif.symbols=c("***"=-.001,"**"=-.01,"*"=-.05)』と初期設定の検定水準の値の前にマイナスを付けてください。



休日・月別・時間帯別のダミー変数を含めた回帰分析 [入力コード11]

```
#lm関数において、説明変数に「+ .」を記入すれば「, data=」の後のデータセットを全て説明変数に含める
#地域の電力利用を気温（3乗項まで）と降水ダミー、休日ダミー、月ダミー
lm_re2<-lm(elwe$elec~elwe$atemp+elwe$atemp2+elwe$atemp3+elwe$wadr_dummy+elwe$off_dummy+. ,data = md)

#地域の電力利用を気温（3乗項まで）と降水ダミー、休日ダミー、時間帯ダミー
lm_re3<-lm(elwe$elec~elwe$atemp+elwe$atemp2+elwe$atemp3+elwe$wadr_dummy+elwe$off_dummy+. ,data = hd)

#地域の電力利用を気温（3乗項まで）と降水ダミー、休日ダミー、月ダミー、時間帯ダミー
lm_re4<-lm(elwe$elec~elwe$atemp+elwe$atemp2+elwe$atemp3+elwe$wadr_dummy+elwe$off_dummy+. ,data = mdhd)

#install.packages("memisc")
library(memisc)

#回帰分析の結果を比較できる形でRの出力に表示
mtable(lm_re1, lm_re2, lm_re3, lm_re4)

#回帰分析の結果を比較できる形でCSVファイルに保存
lm_re1t4<-mtable(lm_re1, lm_re2, lm_re3, lm_re4)
write.mtable(lm_re1t4,file="C:/data/lm_re1t4.csv",colsep=",")
```

日時に関するダミー変数を含めた回帰分析の結果表

◆ 日時に関するダミー変数を含めた回帰分析の結果の出力ファイルの内容を確認します。

● 回帰分析の出力が入ったCSVファイルをExcelなどで開いて、結果を確認します。

- ExcelでCSVファイルを直接開くと、半角括弧内にある各推定係数の標準誤差は負の値と見なされます。この問題を回避するには、Excelの「データ」タブの「テキストファイルの取り込み」から当該CSVファイルを指定し、「カンマ区切り」を設定し、「列のデータ形式」を全て「文字列」として読み込んでください。

回帰分析の結果の比較出力（抜粋）

		気象データのみの 説明変数（基本形）	休日・月ダミーを 説明変数に追加	休日・時間帯ダミーを 説明変数に追加	休日・月・時間帯ダミー を説明変数に追加
		lm_re1	lm_re2	lm_re3	lm_re4
切片	(Intercept)	3453.641*** (26.756)	3567.479*** (26.589)	3666.940*** (15.399)	3575.807*** (15.411)
気温	elwe\$atemp	35.428*** (6.010)	56.734*** (5.845)	-82.870*** (2.832)	-52.765*** (2.890)
気温の2乗項	elwe\$atemp2	-8.527*** (0.391)	-5.532*** (0.389)	-0.953*** (0.184)	-1.670*** (0.189)
気温の3乗項	elwe\$atemp3	0.265*** (0.008)	0.182*** (0.008)	0.115*** (0.004)	0.109*** (0.004)
降水日ダミー	elwe\$wadr_dummy	68.506*** (9.077)	98.521*** (8.286)	43.777*** (4.172)	36.243*** (4.033)
休日（土日祝）ダミー	elwe\$off_dummy		-391.024*** (7.902)	-387.540*** (4.125)	-388.911*** (3.822)
:	:	:	:	:	:
月ダミー（1月基準）	month_2		-70.789** (24.523)		10.826 (11.869)
:	:	:	:	:	:
時間帯ダミー（0時基準）	hour_1			-152.426*** (13.392)	-151.062*** (12.383)
:	:	:	:	:	:
決定係数	R-squared	0.291	0.456	0.851	0.873
自由度調整済み決定係数	adj. R-squared	0.290	0.456	0.851	0.873

複数の回帰分析の式の形で推定係数を確認することで、推定係数の安定性を把握できます。

休日（土日祝）は平日に比べて、約390万kW消費電力が少ないことが示されています。

回帰分析の決定係数を比較すると、どの説明変数（群）の説明力が高いかが分かります。

自由度（＝標本数-説明変数の種類数）が、1万以上などの大きな数であれば「自由度調整済み決定係数」と「決定係数」は僅差になります。

回帰分析の予測値の作成（for関数によるループ）

◆ 4種類の回帰分析の結果に基づいて、各気温に対する消費電力の予測値を作成します。

● 4種類の回帰分析の推定結果を比較するために、各気温における予測値を導出します。

- 1乗項のみの回帰分析では、説明変数の水準に依存せず、各説明変数1単位の増加が被説明変数に同一の効果があると仮定していますが、累乗項を含む回帰分析では、推定係数の直感的な解釈が困難なため、予測値のグラフ等によって効果を可視化して確認します。
- 説明変数として回帰分析に利用したデータに対する予測値は、`predict`関数を利用して『`elwe$pr_lm4 <- predict(lm_re4)`』で簡単に導出することができますが、仮想的な説明変数に対する予測値を導出するため、推定係数と仮想的な説明変数を用いた計算によって予測値を作成します。

● 4種類の回帰分析に基づく予測値を短いプログラムで作成するためにfor関数によるループを利用します。

- プログラムにおいて指定した条件を満たすまで、繰り返し行われる処理をループといいます。Rにおけるfor関数では『`for([変数名] in [変数の下限の自然数] : [変数の上限の自然数])`』と記入すると、中括弧で示されたループを回る度に変数の値が下限から上限まで1ずつ増加していきます。
- 下記のコードでは『`rn`』を変数名として、`coef_re`『`rn:1~4`』という4種の推定係数のベクトルを作成後、0.1℃～35.0℃の気温に対応する予測値の変数`hypo$pr_re`『`rn:1~4`』を作成しています。



回帰分析の結果に基づく4種の予測値の作成 [入力コード12]

仮想的な予測値作成用に350行の新たなデータフレームを作り、行番号に対応する変数lineを入れる

```
hypo <- data.frame("line" = c(1:350))
```

行番号lineに0.1を掛けて0.1℃刻みで気温設定の変数atemp（0.1～35.0℃）を作成

```
hypo$atemp <- hypo$line * 0.1
```

forでrnに1～4までの数字が順次入るループを作り、結果1～4に関する仮想的な気温に基づく予測値を作成

```
for(rn in 1:4){
```

推定結果の1～4からそれぞれの推定係数のベクトル[coef_re番号]を作成

```
eval(parse(text=paste("coef_re",rn,"<-coef(lm_re",rn,")",sep="")))
```

4種の推定係数のベクトルに基づき「（1月平日の0時台）降水なし日」における気温毎の電力消費の予測値を作成

```
eval(parse(text=paste("hypo$pr_re",rn,"<-coef_re",rn,"[1]+coef_re",rn,"[2]*hypo$atemp+  
coef_re",rn,"[3]*hypo$atemp^2+coef_re",rn,"[4]*hypo$atemp^3",sep=""))
```

```
}
```

`paste`関数は、関数内の最後の『`sep=`』に続く引用符内を区切り値として、文字を連結します。この場合は、『`sep=""`』とすることで、区切り値のない文字連結となっています。

for関数で変化させたい値以外の文字列は、引用符で囲って記入します。

`parse`関数では、文字列をRのコマンドとして解釈します。`eval`関数では、Rのコマンドを実行します。

日時ダミーを含めた回帰分析の予測値グラフ

◆ 作成したデータから各気温における消費電力の予測値をグラフに示します。

- `matplot`関数を利用すると、複数の列のデータを用いてグラフを作成することができます。



4種の回帰分析に基づく気温と消費電力の予測値のグラフ作成 [入力コード13]

#気温毎の消費電力予測値のグラフ作成

データフレーム「hypo」の3列目から6列目に入っている4種類の予測値を指定しています。

```
matplot(hypo$atemp,hypo[,3:6], type="l",lwd = 3 , xlab="気温 (°C) " , ylab="消費電力 (万kW) ")
```

```
title("降水のない日における気温と消費電力の関係 (4種の回帰分析の予測値) ")
```

```
legend("topleft", legend=c("lm_re1 (気象データのみ: 基本形) ",
```

```
"lm_re2 (基本形に休日・月ダミーを追加) [平日、1月設定] ",
```

```
"lm_re3 (基本形に休日・時間帯ダミーを追加) [平日、0～1時設定] ",
```

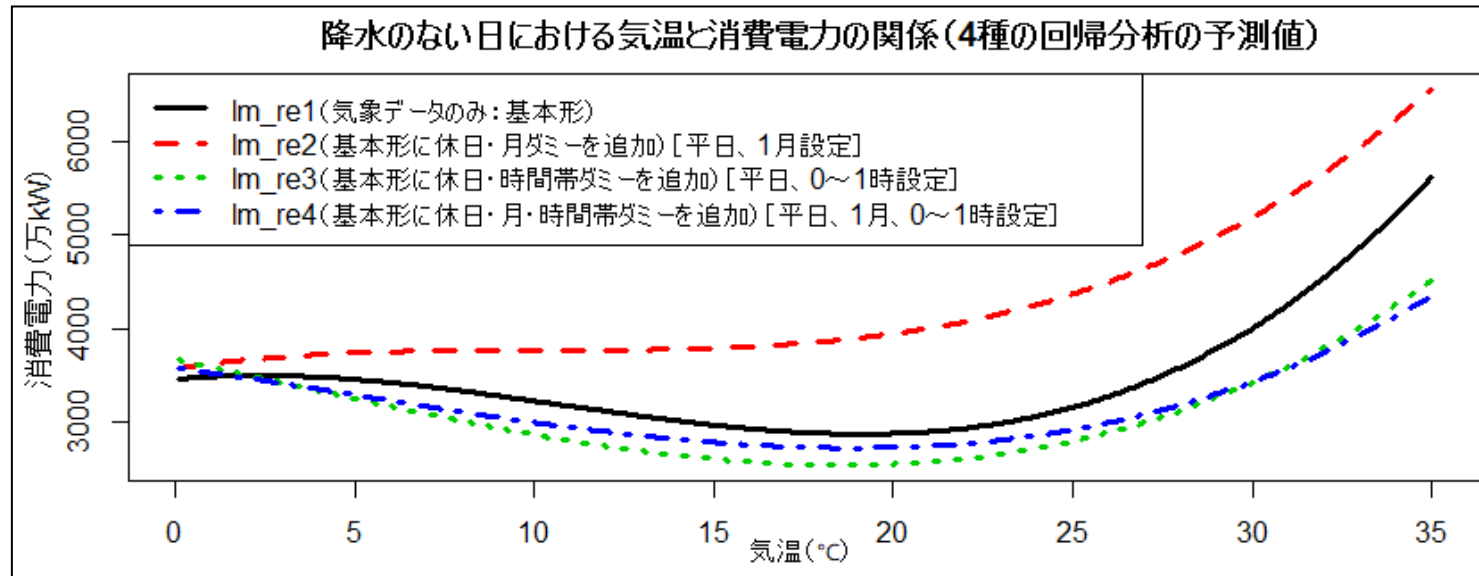
```
"lm_re4 (基本形に休日・月・時間帯ダミーを追加) [平日、1月、0～1時設定] ")
```

```
,col=1:4,lty=1:4,lwd = 3, text.width=25)
```

『type="l"』では、線によるグラフ作成を指定し、『lwd = 3』では線の太さを指定しています。

いったんグラフを作成した後に日本語でグラフのタイトル (title) と凡例 (legend) を追記します。

Rにおける4種の回帰分析に基づく気温と消費電力の予測値のグラフ



回帰分析の結果に基づく測定・予測・発見

◆ 回帰分析の結果に基づいて定量的な効果の測定、予測、発見を行うことができます。

- 分析結果に基づいて、説明変数の効果を定量的に測定することができます。
 - ・ 休日が平日に比べて消費電力が小さいことは経験的にも認知されていますが、回帰分析を行うことで東京電力管内では約390万kW小さくなることが定量的に測定できます。
- 回帰分析の推定結果を用いれば、月・時間帯の設定、天気予報における消費電力の予測を行うことができます。
 - ・ 回帰分析 (`lm_re4`) で得られた「気象データ、休日・月・時間帯ダミー」の結果を用いれば、「7月の平日14時において予測気温28.5℃、晴れ予報」といった任意の月・時間帯における天気予報に基づく消費電力の予測を行うことができます。
- 分析結果から、特定の条件を満たす値を導出したり、**新たな発見**が得られたりすることがあります
 - ・ `which.min`関数では、最小値の変数があるデータフレームの位置を示すことができるため、変数の最小値やそれをもたらす条件を表示できます。回帰分析 (`lm_re4`) に基づけば、東京電力管内において、最小の消費電力をもたらす気温は18.8℃であることが示されます。
 - ・ 東京電力管内と同様の最小消費電力をもたらす気温を9つの電力管内で導出すると、日本の北東側で低く、南西側で高い傾向にあることが分かります。
 - ・ コードの記述を少し改変するだけで、すぐに同種の処理・分析が行えることもプログラミングの利点の一つです。



休日・月・時間帯ダミーを追加した回帰分析で最小の消費電力をもたらす気温導出 [入力コード14]

#lm_re4 (基本形に休日・月・時間帯ダミーを追加) の予測値に基づく消費電力の最小値をもたらす気温

`hypo[which.min(hypo$pr_re4),2]` ← データフレームhypoにおける「hypo\$pr_re4」の最小値の行、2列目（気温の列）を表示します。

休日・月・時間帯ダミーを追加した回帰分析で最小の消費電力をもたらす気温

```
> hypo[which.min(hypo$pr_re4),2]
[1] 18.8
```

北海道電力	16.0℃	中部電力	19.7℃	中国電力	20.2℃
東北電力	17.5℃	北陸電力	18.3℃	四国電力	19.8℃
東京電力	18.8℃	関西電力	19.6℃	九州電力	19.4℃

- 気温や日時データの他にも、追加を検討すべき説明変数として「電気代」が考えられます。本教材の【参考2】ではe-Stat APIから、「電力代」に関する物価指数を取得する手順を示しています。

【参考1】RとRStudioのダウンロード

◆RとRStudioは、誰でもウェブサイトからダウンロードすることができます。

- Rのインストール用ファイルは、**CRAN**に参加する統計数理研究所のウェブサイトからダウンロードすることができます。

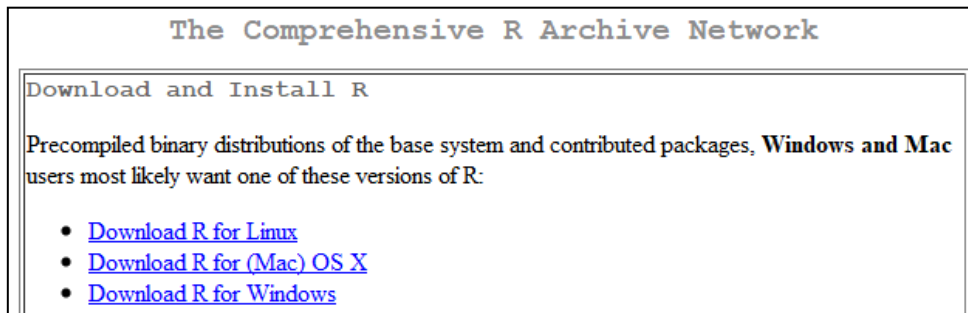
<https://cran.ism.ac.jp/>

What are **R** and **CRAN**?



- CRAN（シーラン：Comprehensive R Archive Network）は、Rに関するファイルを蓄積・提供する国際ネットワークです。
- 2018年3月時点における上記URLのウェブサイトの表記は概ね英語ですが、英単語が分かれば、ダウンロードやインストールに支障はありません。
- OSへインストールするためのRには、Windows版、Macintosh版、Linux版がありますが、この講座ではWindows版で説明します。
- Windowsを利用している場合は「Download R for Windows」をクリックした後に表示されるWindows版のダウンロードボタンをクリックしてください。

OSに応じたRの選択画面



Windows用Rのダウンロード画面

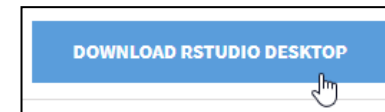


- RStudioのインストール用ファイルは、RStudioのウェブサイトからダウンロードできます。

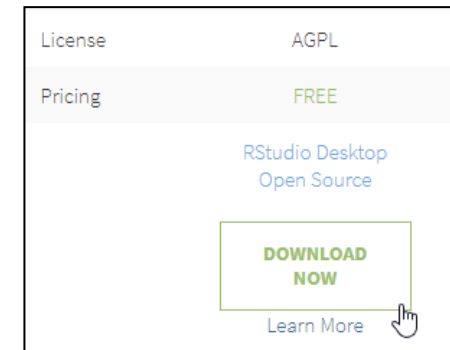
<https://www.rstudio.com/products/rstudio/download/>

- RStudioには、各PCの中のRを実行するデスクトップ版と離れたサーバ上のRを実行するサーバ版がありますが、一般にはデスクトップ版を利用します。
- RStudioのトップページからの移動する場合は、まず画面上部の「Products> RStudio」を選択してください。次に表示される画面で「Open Source Edition」の欄にある「DOWNLOAD RSTUDIO DESKTOP」のボタンを押します。続いて表示される画面でオープンソース版の「DOWNLOAD NOW」をクリックしてください。

デスクトップ版の
ダウンロードへのリンク



オープンソース版の
ダウンロードボタン

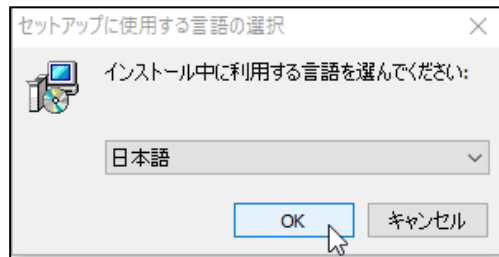


【参考1】RとRStudioのインストール

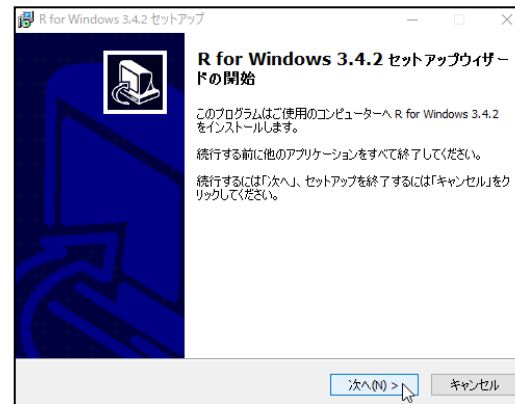
◆RとRStudioは、マウスによる操作だけで簡単にインストールすることができます。

- Rのインストールにおいては、全て初期設定で「OK」や「次へ」で進めて、問題ありません。
 - ・ インストール時の初期設定が把握でき、変更したい方は、インストール先のフォルダの指定、32bit版か64bit版等の選択をしてください。設定内容が把握できない方や細かい設定を気にしない方は、全て初期設定でのインストール、32bit版と64bit版の両方のインストールで構いません。

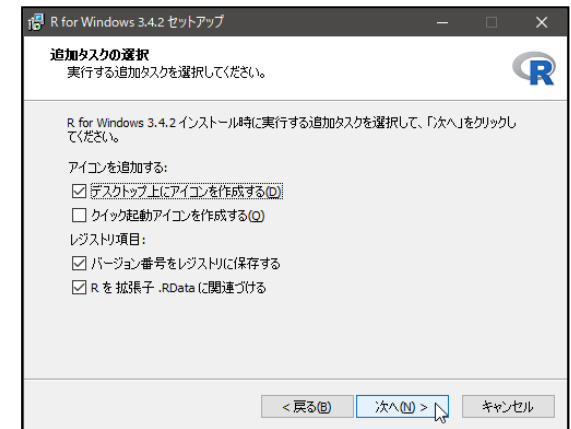
Rのインストールの言語選択



Rのインストール開始画面



Rのインストール時の最後の選択



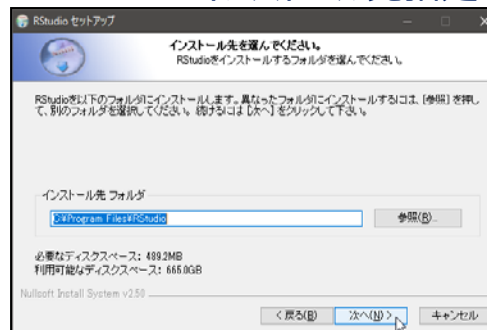
- RStudioのインストールも、全て初期設定で「次へ」で進めて、問題ありません。

- ・ 初期設定でインストールを完了すると、スタートメニューの中にRStudioのショートカットができます。

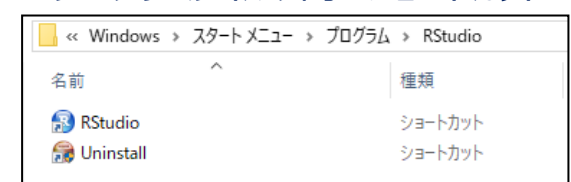
RStudioインストール開始



RStudioのインストール先指定



プログラムフォルダ内のショートカット



【参考2】e-Statへのユーザー登録

◆ e-Stat APIを利用するために、まずe-Statにユーザー登録を行います。

- e-Statウェブサイトの「新規登録」または下記のURLからユーザー登録を行います。

<https://www.e-stat.go.jp/mypage/user/preregister>

- ユーザーIDとなるE-mailアドレスを登録して、e-Statへの仮登録を行ってください。

- 仮登録画面でメールアドレスを記入すると、そのメールアドレスに対してe-Statから自動送信メールが送付されます。そのメールの本文内にあるURLから本登録用のウェブページを開いてください。



e-StatへのユーザーID（E-mailアドレス）の仮登録欄

ユーザーIDを入力し、「仮登録」ボタンをクリックしてください。

ユーザーID（E-mailアドレス）（必須）	<input type="text"/>
------------------------	----------------------

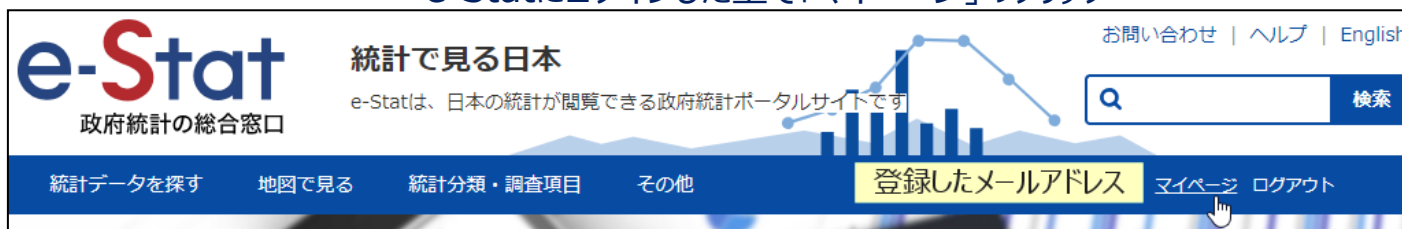
- 本登録においては、利用する機能に「API機能」を含めて指定し、ログイン時に利用するパスワードを指定してください。

e-Statのユーザー登録において利用する機能の選択

利用する機能（必須）	<input checked="" type="checkbox"/> e-Stat <input checked="" type="checkbox"/> API機能 <input checked="" type="checkbox"/> 地図で見る統計(jSTATMAP)	利用したい機能にチェックを入れてください。
------------	--	-----------------------

- 本登録の完了後e-Statにログインし、E-mailアドレス表示の右側にある「マイページ」をクリックしてください。

e-Statにログインした上で「マイページ」のクリック

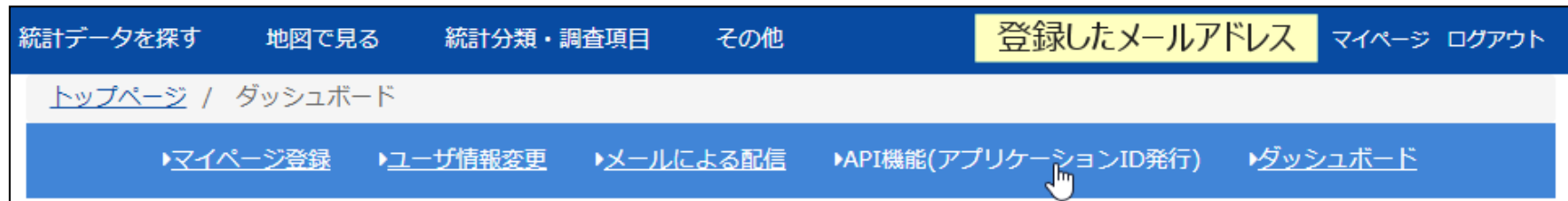


【参考2】e-Stat APIのアプリケーションID発行

◆e-Stat APIを利用するためのアプリケーションIDの発行を行います。

- マイページの上部に表示される「API機能（アプリケーションID発行）」をクリックしてください。

e-Statの「マイページ」内の「API機能（アプリケーションID発行）」のクリック



- 「名称」および「URL」を記入して、「発行」ボタンをクリックして、e-Stat APIのアプリケーションIDを発行します。
 - URLは、ウェブサービスからe-Stat APIを利用する場合は当該ウェブサービスのURLを入力します。今回のようにRのプログラミングによってPCから利用したり、非公開サイトでの利用を行う場合は、「http://localhost/」と入力してください。

e-Statにおける「API機能（アプリケーションID発行）」の発行と表示

1	*名称	総務省ICTスキル総合習得プログラム 講座4-3に基づくRでの利用	<div>発行</div> <div>変更</div> <div>廃止</div>
	*URL	http://localhost/	
	概要	講座4-3【参考2】に基づく試行	
	appId	発行されたアプリケーションID	

- プログラミングでは、表示されたアプリケーションIDを含めてコードを書き、e-Stat APIに情報提供を要求します。

- e-Stat APIのアプリケーションIDは英数字の羅列で構成されています。
- e-Stat APIのアプリケーションIDは、各利用者の各サービスに対応する形で発行されています。アプリケーションIDを他者に教えたり、公開したりすることがないように取り扱いには注意してください。

【参考2】Rにおけるestatapiパッケージの利用

◆ Rではe-StatのAPIを簡単に利用できるestatapiパッケージを利用することができます。

- Rにおけるestatapiパッケージを利用して、e-Stat内で『消費者物価指数』が含まれる統計を検索します。
 - e-Statは総務省統計局が運営している国のサービスですが、Rのestatapiパッケージは有志の個人が作成したものです。estatapiパッケージの詳細な利用方法は、CRAN内の説明ファイル等を参照してください。（<http://cran.r-project.org/pub/R/web/packages/estatapi/README.html>）
 - e-StatのAPIを利用するプログラミングコードとして、実習用のデータ・コードに含まれる「code_ref2_estat_api.R」を用いて説明します。
 - 本教材における消費電力ビッグデータの分析と関連して、インターネットを経由して消費者物価指数の「電気代」のデータを取得することを想定します。



estatapiパッケージの利用宣言と統計名の検索 [入力コード15]

#estatapiパッケージのダウンロードと利用宣言

```
#install.packages("estatapi")
```

```
library(estatapi)
```

```
yourID <- "各利用者が取得したe-Stat APIのアプリケーションIDを記入"
```

#e-Stat内を『消費者物価指数』で検索

```
estat_getStatsList(appId = yourID, searchWord = "消費者物価指数")
```

#データ形式tibbleにおいて、表示幅を制限しない設定

```
options(tibble.width = Inf)
```

#データ形式tibbleにおいて、全ての行を表示する設定

```
options(tibble.print_max = Inf)
```

- 実行結果として、『消費者物価指数』が含まれる公的統計が下記のように表示されます。

Rの出力における『消費者物価指数』が含まれる統計情報の表示

```
> #e-Stat内を『消費者物価指数』で検索
> estat_getStatsList(appId = yourID, searchWord = "消費者物価指数")
# A tibble: 4 x 13
  `@id`      STAT_NAME      GOV_ORG STATISTICS_NAME
  <chr>      <chr>          <chr>    <chr>
1 0000010212 社会・人口統計体系 総務省   都道府県データ 社会生活統計指標
2 0002050001 消費者物価指数     総務省   平成17年基準消費者物価指数
3 0003036792 消費者物価指数     総務省   平成22年基準消費者物価指数
4 0003143513 消費者物価指数     総務省   2015年基準消費者物価指数
```

この枠内のコードは記入しなくても、データの取得は可能ですが、Rの出力から各種IDを正確に確認するために必要な設定です。

取得対象とする『2015年基準消費者物価指数』の統計IDは「003143513」であることを確認します。

- 本教材においては、統計IDが「003143513」の『2015年基準消費者物価指数』を採りあげます。

【参考2】estatapiパッケージを用いたコードの確認

◆ estatapiパッケージを用いて、「電気代」の地域別物価指数のコードを確認します。

- 統計IDが「003143513」の『2015年基準消費者物価指数』を指定し、データ説明を「data_info」に格納します。
- RStudioの環境内の「data_info」をクリックし、[cat01]に品目コード、[area]に地域コードがあることを特定します。
- [cat01]と[area]を改めてRのデータに格納し、環境から変数をダブルクリックし、データビューでコードを確認します。

 データ説明の格納と収集対象とするコードの確認 [入力コード16]

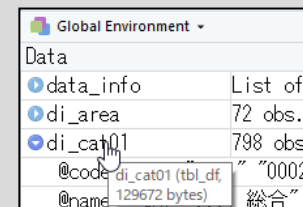
```
# 2015年基準消費者物価指数 (statsDataId=0003143513) のデータ説明をdata_infoへ格納
data_info <- estat_getMetaInfo(appId=yourID, statsDataId="0003143513")
```

```
# 品目コードと地域コードを調べるため、cat01およびareaをRのデータフレームとして格納
```

```
di_cat01<-data_info$cat01
```

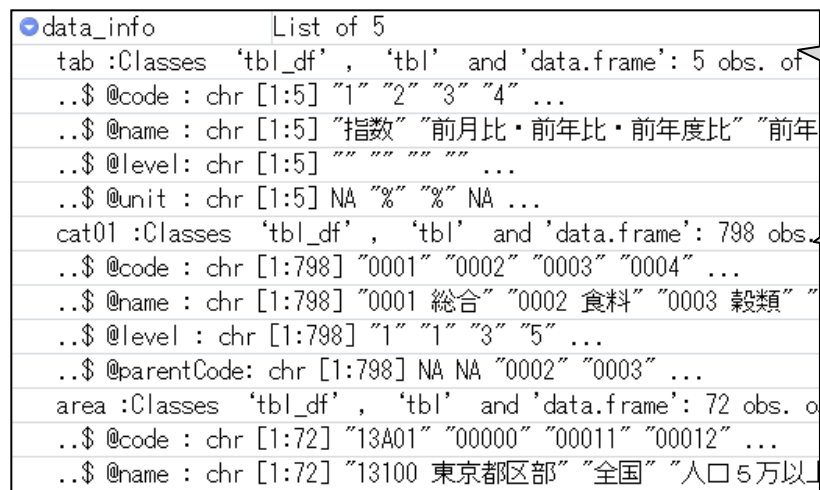
```
di_area<-data_info$area
```

環境（ワークスペース）内の変数をダブルクリックして、データビューに表示します。



Variable	Type	Size
data_info	List of	
di_area	tbl_df	72 obs.
di_cat01	tbl_df	798 obs.
@code	di_cat01 (tbl_df)	"0002"
@name	129672 bytes	総合

RStudioの環境（ワークスペース）内の表示



Variable	Type	Size
data_info	List of 5	
tab	:Classes 'tbl_df', 'tbl' and 'data.frame': 5 obs. of	
cat01	:Classes 'tbl_df', 'tbl' and 'data.frame': 798 obs.	
area	:Classes 'tbl_df', 'tbl' and 'data.frame': 72 obs. of	

物価指数の種類の変数名は [tab] です。

対象品目の変数名は [cat01] です。

対象地域の変数名は [area] です。

データビューでのコードの確認

	@code	@name
307	3185	3185 壁紙張替費
308	3180	3180 火災・地震保険料
309	0054	0054 光熱・水道
310	0056	0056 電気代

[di_cat01] より「電気代」の品目コード [0056] を確認します。

	@code	@name
22	01A01	01100 札幌市
23	02A01	02201 青森市
24	03A01	03201 盛岡市
25	04A01	04100 仙台市

[di_area] より「札幌市」の地域コード [01A01] を確認します。

【参考2】estatapiパッケージを用いたデータ取得

◆ estatapiパッケージを用いて、指定地域の「電気代」の物価指数を取得します。

- 「統計ID」「品目コード」「地域コード」等を指定して、各地域の「電気代」物価指数をepdataに格納します。
 - APIで指定可能なパラメーターは、e-Stat APIの仕様 (<https://www.e-stat.go.jp/api/e-stat-manual>) を確認してください。



コード指定によるデータを取得とCSV保存 [入力コード17]

```
#取得したコードを指定してe-Stat APIからデータを取得し、epdataに格納
epdata<-estat_getStatsData(appId = yourID,
  statsDataId = "0003143513", #2015年基準消費者物価指数
  cdTab="1", #物価指数（データ説明の変数名に接頭語としてcdを付け、変数名の最初を大文字で記入）
  cdCat01 = "0056", #電気代
  cdArea = c("01A01","04A01","13A01","23A01","16A01","27A01","34A01","37A01","40A02","47A01"),
  #札幌市、仙台市、…、福岡市の9地域指定
  lvTime = "4", #月次データ
  cdTimeFrom="2016") #データ取得開始年

#epdataをCSVに出力
write.csv(epdata, "C:/data/estat_elec_9area_price.csv")
```

あらかじめ、Cの直下に「data」というフォルダを作っておけば、その中にCSVファイル「estat_elec_9area_price.csv」が出力されます。

- epdataには指定した地域・時期のデータが格納されるとともに、データをCSVファイルに保存しています。

RStudioのデータビューにおけるepdataの表示

tab_code	表示項目	cat01_code	2015年基準品目	area_code	地域(2015年基準)	time_code	時間軸(年・月)	value
1	指数	0056	0056 電気代	01A01	01100 札幌市	2018000202	2018年2月	101.7
1	指数	0056	0056 電気代	01A01	01100 札幌市	2018000101	2018年1月	100.9

- APIを利用すると、様々なデータソースから対象データを絞って収集することが可能となります。