

2016/8/9 AJACSこまち

次世代シーケンサーと統計解析

岩手医科大学
いわて東北メディカル・メガバンク機構
生体情報解析部門 助教
古川 亮平

本日のコンテンツ

- Rの基本操作
- 統計量を算出して表にまとめる
- 回帰分析／分散分析を行い、分析結果から情報を取り出してまとめる
- 多重性の問題を補正したP値で回避する

膨大なデータの中から、目的のデータにうまくアクセスし、個々の解析結果からうまく情報を取り出してまとめ、目的に沿ったリストを得る。



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



[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Mailing Lists](#)

[Reporting Bugs](#)

[Development Site](#)

[Conferences](#)

[Search](#)

R Foundation

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

Documentation

[Manuals](#)

[FAQs](#)

[The R Journal](#)

[Books](#)

[Certification](#)

[Other](#)

Links

[Bioconductor](#)

[Related Projects](#)

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. [To download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

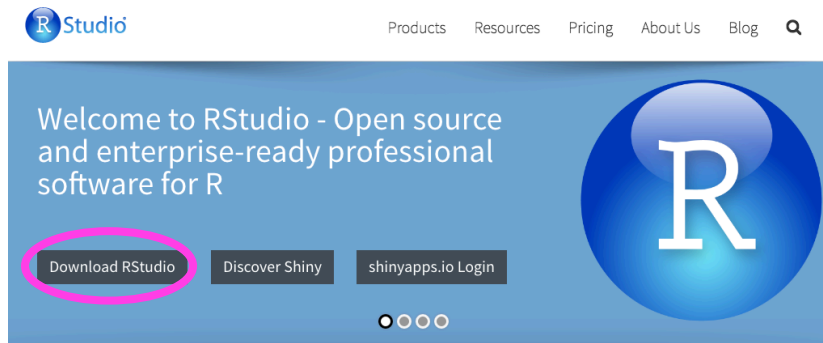
News

- The **useR!** 2017 conference will take place in Brussels, July 4 - 7, 2017, and details will be appear here in due course.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, has taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.


- 統計処理言語「S」を基に、フリーソフトとして開発された統計解析環境・プログラミング言語。
- コマンドラインインターフェースが基本。スクリプトを書いて一括処理を行うことも可能。
- ベクトル・行列演算が簡便かつ効率的に行える。
- 独自の関数を作成することで機能拡張できる。
- 作成した関数等をパッケージ単位でまとめることができる。様々なパッケージを導入することで、最先端の統計手法を用いることができる。

RStudio

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




- Rのための統合開発環境。
- 直感的なユーザーインターフェースにより、Rによる作業が楽になる。



Powerful IDE for R

RStudio IDE is a powerful and productive user interface for R. It's free and open source, and works great on Windows, Mac, and Linux.


[Learn More >](#)



R Packages

Our developers and expert trainers are the authors of several popular R packages, including ggplot2, plyr, lubridate, and others.

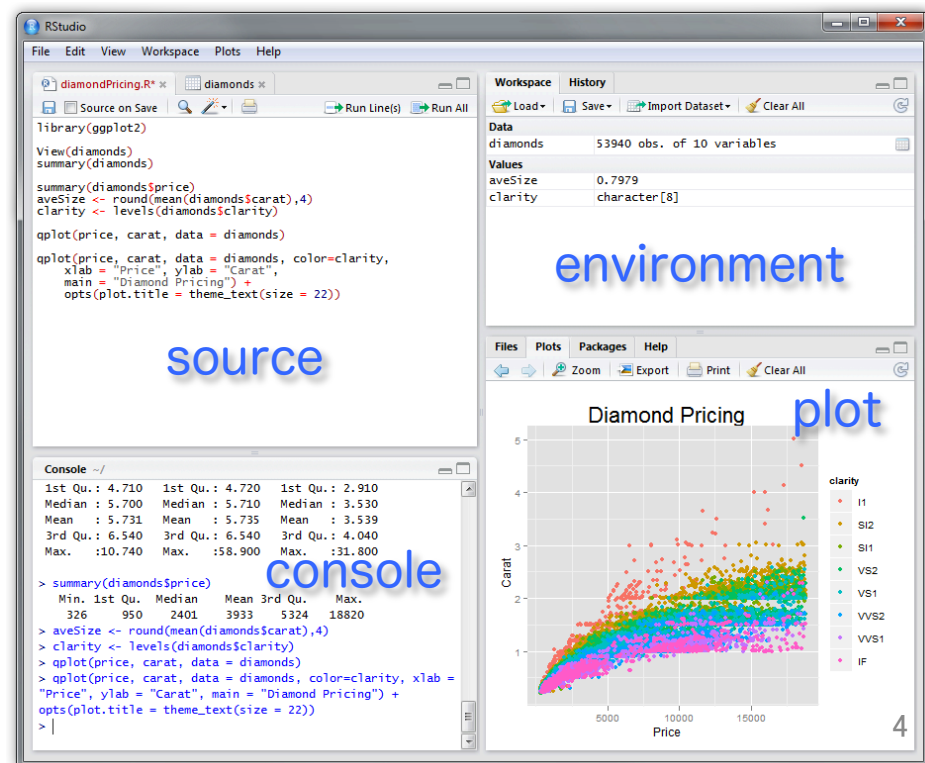
[Learn More >](#)



Bring R to the web

Shiny is an elegant and powerful web framework for building interactive reports and visualizations using R — with or without web development skills.

[Learn More >](#)



演算

算術演算子

+	加算
-	減算
*	乗算
/	除算
%%	整数除算
%%	剰余
^	累乗

```
> 2+3
```

```
[1] 5
```

```
> 2*3
```

```
[1] 6
```

```
> 5/2
```

```
[1] 2.5
```

```
> 2^4+3
```

```
[1] 19
```

```
> 2^(4+3)
```

```
[1] 128
```

```
> exp(4)
```

```
[1] 54.59815
```

```
> pi-3
```

```
[1] 0.1415927
```

```
> log(4)
```

```
[1] 1.386294
```

```
> log2(4)
```

```
[1] 2
```

```
> log10(4)
```

```
[1] 0.60206
```

```
> 0.05/1E5
```

```
[1] 5e-07
```

```
> 0.05E6
```

```
[1] 50000
```

```
> 5%%2
```

```
[1] 2
```

```
> 5%%2
```

```
[1] 1
```

変数と代入

- 計算結果を変数に代入することにより、結果を再利用できる。
- 変数名にはアルファベット、数字、'.'（ドット）、'_'（アンダーバー）が利用可能。
- 先頭はアルファベットかドット。
- 大文字と小文字は区別される。
- 代入を表す演算子には、<-、=、->の3通りがある。

例1：以下はいずれもaに4が代入される

```
> a <- 1 + 3
```

```
> a = 1 + 3
```

```
> 1 + 3 -> a
```

例2：代入したaを利用した演算

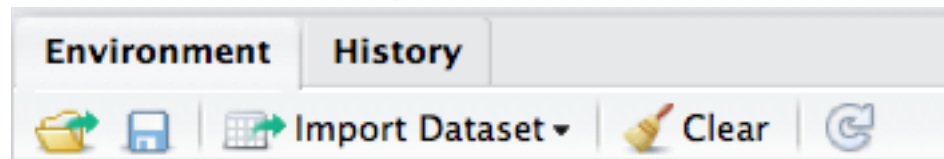
```
> b <- 7
```

```
> a + b
```

```
[1] 11
```

ヒストリー（履歴）

- コンソール上で、上下矢印キー（↑ ↓）により、コマンドの履歴を前後にたどることができる。
- 左右矢印キー（← →）により、カーソルを左右に動かすことができる。これによって、コマンドの編集ができる。
- ヒストリータブを使って履歴をたどることも可能。



ベクトルの作成 1

```
> #ベクトルは関数cを用いて作成する
```

```
> a <- c(1,4,6,3,9)
```

```
> a
```

```
[1] 1 4 6 3 9
```

```
> #1から5までの連続した整数
```

```
> b <- 1:5
```

```
> b
```

```
[1] 1 2 3 4 5
```

```
> #3から-3まで1ずつ減少するベクトル
```

```
> c <- c(3:-3)
```

```
> c
```

```
[1] 3 2 1 0 -1 -2 -3
```

```
> #ベクトルの長さ (要素数)
```

```
> length(a)
```

```
[1] 5
```

```
> #bの4番目の要素
```

```
> b[4]
```

```
[1] 4
```


ベクトルの作成2

```
> #規則的なベクトルの作成
> d <- rep(c(1,2,4),times=2)
> d
[1] 1 2 4 1 2 4
> e <- rep(c(1,2,4),each=2)
> e
[1] 1 1 2 2 4 4
> f <- rep(c("Dog","Cat","Mouse"),times=c(2,3,5))
> f
[1] "Dog" "Dog" "Cat" "Cat" "Cat" "Mouse" "Mouse" "Mouse" "Mouse" "Mouse"
> g <- seq(0,10,by=2)
> g
[1] 0 2 4 6 8 10
> h <- seq(0,10,length=5)
> h
[1] 0.0 2.5 5.0 7.5 10.0
```

ベクトルの演算

```
> #a <- c(1,4,6,3,9)、b <- 1:5と設定されている
```

```
> 1 + a
```

```
[1] 2 5 7 4 10
```

```
> 3 * a
```

```
[1] 3 12 18 9 27
```

```
> a + b
```

```
[1] 2 6 9 7 14
```

```
> a * b
```

```
[1] 1 8 18 12 45
```

```
> a > 3
```

```
[1] FALSE TRUE TRUE FALSE TRUE
```

```
> b < 3
```

```
[1] TRUE TRUE FALSE FALSE FALSE
```

```
> a < b
```

```
[1] FALSE FALSE FALSE TRUE FALSE
```

```
> a >= 2 & a < 5
```

```
[1] FALSE TRUE FALSE TRUE FALSE
```

```
> a >= 2 & a%%2!=0
```

```
[1] FALSE FALSE FALSE TRUE TRUE
```

比較演算子

>,<,>=,<=	不等号
==	等しい
!=	等しくない

論理演算子

&	論理積「且つ」
	論理和「または」
!a	aの否定

ベクトルの要素の抽出

```
> #a <- c(1,4,6,3,9)と設定されている
```

```
> a[2:4]
```

```
[1] 4 6 3
```

```
> a[c(2,4,1)]
```

```
[1] 4 3 1
```

```
> a[c(T,T,F,F,T)]
```

```
[1] 1 4 9
```

```
> a[a>3]
```

```
[1] 4 6 9
```

```
> a[a>=2 & a%%2!=0]
```





```
[1] 3 9
```

オブジェクト

- コンソール上でls()と入力すると、変数に保存されたデータ（オブジェクト）のリストを参照できる。

```
> ls()  
[1] "a" "b" "c" "d" "e" "f" "g" "h"
```

- Environmentタブで、より詳細な情報を閲覧可能。

Environment		History
  Import Dataset ▾  Clear 		
Global Environment ▾		
Values		
a	num [1:5]	1 4 6 3 9
b	int [1:5]	1 2 3 4 5
c	int [1:7]	3 2 1 0 -1 -2 -3
d	num [1:6]	1 2 4 1 2 4
e	num [1:6]	1 1 2 2 4 4
f	chr [1:10]	"Dog" "Dog" "Cat" "Cat" "Cat" "Mouse" "Mouse" "Mouse" "Mouse" "Mouse"
g	num [1:6]	0 2 4 6 8 10
h	num [1:5]	0 2.5 5 7.5 10

- rm(変数名)で、オブジェクトを消去できる。

関数

- 関数名(引数1, 引数2, ...)
- 関数は、一般に複数の引数を入力として何らかの計算を行い、1つのオブジェクトを返す。
- 引数には必須のものと省略可能なものがあり、後者は省略するとデフォルト値が適用される。

例1：ベクトルaを小さい順（increasing order）でソートする

```
> sort(a)  
[1] 1 3 4 6 9
```

例2：ベクトルaを小さい順（decreasing order）でソートする

```
> sort(a,decreasing=T)  
[1] 9 6 4 3 1
```

- 関数のマニュアルは、help(関数名)、または?関数名で表示できる。

基本統計量の計算

```
> #a <- c(1,4,6,3,9)と設定されている
> length(a)
[1] 5
> sum(a)
[1] 23
> mean(a)
[1] 4.6
> median(a)
[1] 4
> min(a)
[1] 1
> max(a)
[1] 9
> range(a)
[1] 1 9
> var(a)
[1] 9.3
> sd(a)
[1] 3.04959
> summary(a)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.0    3.0    4.0    4.6    6.0    9.0
```

行列の作成

> #1つのベクトルを行列の形に並べる

> m1 <- matrix(1:9,3,3)

> m1

```
      [,1] [,2] [,3]  
[1,]    1    4    7  
[2,]    2    5    8  
[3,]    3    6    9
```

> #複数のベクトルを行、または列として並べる

> A <- c(1,4,6)

> B <- c(2,3,9)

> C <- c(4,6,5)

> m2 <- cbind(A,B,C)

> m2

```
      A B C  
[1,] 1 2 4  
[2,] 4 3 6  
[3,] 6 9 5
```

> m3 <- rbind(A,B,C)

> m3

```
      [,1] [,2] [,3]  
A         1     4     6  
B         2     3     9  
C         4     6     5
```

行列の要素の取り出し

```
> m1
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> #m1の2行3列目の要素
> m1[2,3]
[1] 8
> #m1の2行目の要素
> m1[2,]
[1] 2 5 8
> #m1の3列目の要素
> m1[,3]
[1] 7 8 9
> #1,3行目、2,3列目の要素からなる部分行列
> m1[c(1,3),2:3]
      [,1] [,2]
[1,]    4    7
[2,]    6    9
> #行列の大きさ
> dim(m1)
[1] 3 3
```


行列の演算

```
> m1
```

```
      [,1] [,2] [,3]  
[1,]    1    4    7  
[2,]    2    5    8  
[3,]    3    6    9
```

```
> m2
```

```
      A B C  
[1,] 1 2 4  
[2,] 4 3 6  
[3,] 6 9 5
```

```
> #要素ごとの足し算
```

```
> m1 + m2
```

```
      A B C  
[1,] 2 6 11  
[2,] 6 8 14  
[3,] 9 15 14
```

```
> #要素ごとのかけ算
```

```
> m1 * m2
```

```
      A B C  
[1,] 1 8 28  
[2,] 8 15 48  
[3,] 18 54 45
```

```
> #行列積
```

```
> m1 %*% m2
```

```
      A B C  
[1,] 59 77 63  
[2,] 70 91 78  
[3,] 81 105 93
```

```
> #転置行列（行と列の入れ替え）
```

```
> t(m1)
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6  
[3,]    7    8    9
```

作業ディレクトリの設定

- 作業ディレクトリ：読み込むデータファイルや結果を書き出すファイルを保存するディレクトリ。
- `getwd()`で現在の作業ディレクトリを確認できる。
- `setwd("ディレクトリ名")`で設定できる。
- ディレクトリの設定は必須ではないが、設定しない場合、ファイルの読み込みや保存の際にディレクトリを指定する必要がある。

例：デスクトップにあるpracticeフォルダ内のdata.txtを読み込む場合

```
> setwd("~/Desktop/practice")  
> data <- read.table("data.txt",header=T)  
or  
> data <- read.table("~/Desktop/practice/data.txt",header=T)
```

データフレーム

- Rの統計解析で最も基本的なデータ。
- 行列のような表形式のデータだが、各列が異なる型のデータを持ちうる（行列は全てが同じ型のデータ）。
 - ✓ Rのデータ型として以下のものがある。
 - ◇ 数値型：numeric（実数または整数）
 - ◇ 論理型：logical（TRUE/FALSE）
 - ◇ 文字列型：character（“cg00000029”, “promoter”などのように、二重引用符で囲まれた文字列として表す）

CpG	Gene	region	P.value	r.squared
cg00000029	RBL2	promoter	9.352687e-01	3.066766e-04
cg00000108	C3orf35	last-exon	5.372680e-02	1.588396e-01
cg00000109	FNDC3B	intron	3.100466e-01	4.678213e-02
cg00000236	VDAC3	last-exon	1.276800e-01	1.022629e-01
cg00000289	ACTN1	last-exon	8.820903e-01	1.022398e-03
cg00000292	ATP2A1	first-exon	8.216525e-01	2.360013e-03
cg00000363	PGBD5	intron	4.104474e-01	3.101040e-02

データフレームの読み込み

- read.tableが読み込みのための基本的な関数
- デフォルト値の異なる読み込み関数がいくつか用意されている。

read.table(file, options…)

デフォルトでセパレータは空白文字、header=FALSE

read.delim(file, options…)

デフォルトでセパレータはタブ、header=TRUE

read.csv(file, options…)

デフォルトでセパレータはカンマ、header=TRUE

```
> MEA.data <- read.table("MEA.data.txt",header=T)
```

```
> head(MEA.data)
```

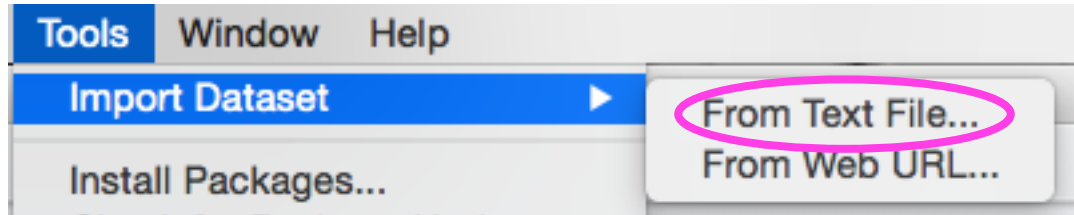
	CpG	Gene	region	P.value	r.squared
1	cg00000029	RBL2	promoter	0.9352687	0.0003066766
2	cg00000108	C3orf35	last-exon	0.0537268	0.1588396212
3	cg00000109	FNDC3B	intron	0.3100466	0.0467821335
4	cg00000236	VDAC3	last-exon	0.1276800	0.1022628986
5	cg00000289	ACTN1	last-exon	0.8820903	0.0010223983
6	cg00000292	ATP2A1	first-exon	0.8216525	0.0023600131

```
> dim(MEA.data)
```

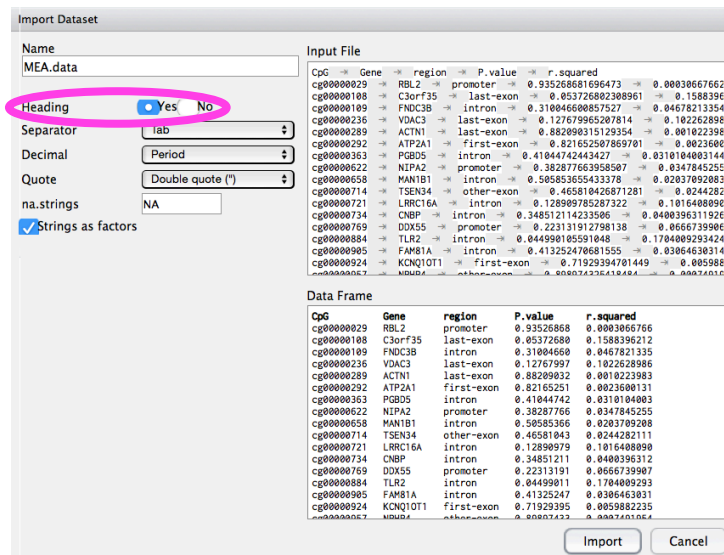
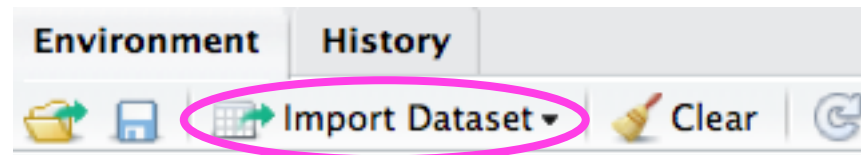
```
[1] 276663      5
```

nrow(MEA.data) : 行数
ncol(MEA.data) : 列数

データフレームの読み込み



または



```
> MEA.data <- read.delim("~/Desktop/MEA.data.txt")
> View(MEA.data)
```

CpG	Gene	region	P.value	r.squared
cg00000029	RBL2	promoter	9.352687e-01	3.066766e-04
cg00000108	C3orf35	last-exon	5.372680e-02	1.588396e-01
cg00000109	FNDC3B	intron	3.100466e-01	4.678213e-02
cg00000236	VDAC3	last-exon	1.276800e-01	1.022629e-01
cg00000289	ACTN1	last-exon	8.820903e-01	1.022398e-03
cg00000292	ATP2A1	first-exon	8.216525e-01	2.360013e-03

データフレームの情報表示

```
> str(MEA.data) #各列のデータ型と内容の一部を表示
```

```
'data.frame': 276663 obs. of 5 variables:
 $ CpG      : Factor w/ 276663 levels "cg00000029","cg00000108",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Gene      : Factor w/ 16092 levels "A1BG","A1BG-AS1",...: 11400 1767 4998 15046 176 1009 10207 9334 8180 14567 ...
 $ region    : Factor w/ 5 levels "first-exon","intron",...: 5 3 2 3 3 1 2 5 2 4 ...
 $ P.value   : num  0.9353 0.0537 0.31 0.1277 0.8821 ...
 $ r.squared: num  0.000307 0.15884 0.046782 0.102263 0.001022 ...
```

```
>
```

```
> summary(MEA.data) #各列のデータの概要を表示
```

	CpG		Gene		region		P.value		r.squared
cg00000029:	1	PTPRN2 :	1202	first-exon:	37458	Min. :	0.0000001	Min. :	0.00000
cg00000108:	1	MAD1L1 :	695	intron :	140606	1st Qu.:	0.2565939	1st Qu.:	0.00458
cg00000109:	1	PRDM16 :	602	last-exon :	21765	Median :	0.5059138	Median :	0.02037
cg00000236:	1	TNXB :	540	other-exon:	21608	Mean :	0.5042426	Mean :	0.04275
cg00000289:	1	DIP2C :	465	promoter :	55226	3rd Qu.:	0.7533672	3rd Qu.:	0.05808
cg00000292:	1	RPTOR :	416			Max. :	0.9999947	Max. :	0.74773
(Other)	:276657	(Other):	272743						

因子 factor

- 限られた数のカテゴリで表されるもの。
- 因子の取り得る値を水準 level という。
例) region列は5つの水準を持つ。

```
> levels(MEA.data$region)
```

```
[1] "first-exon" "intron"      "last-exon"   "other-exon"  "promoter"
```

データフレームの要素の取り出し

- データフレームの要素の取り出しは、行列の要素の取り出しと同じ！
(p16参照)
- 練習問題
MEAデータから、CpG列、Gene列を取り出して、annotationに代入する。

```
> annotation <- MEA.data[,1:2]  
> head(annotation)
```

	CpG	Gene
1	cg00000029	RBL2
2	cg00000108	C3orf35
3	cg00000109	FNDC3B
4	cg00000236	VDAC3
5	cg00000289	ACTN1
6	cg00000292	ATP2A1

```
> anotation <- MEA.data[,c("CpG","Gene")]  
> head(anotation)
```

	CpG	Gene
1	cg00000029	RBL2
2	cg00000108	C3orf35
3	cg00000109	FNDC3B
4	cg00000236	VDAC3
5	cg00000289	ACTN1
6	cg00000292	ATP2A1

条件式による部分データの抽出1

`subset(data, 条件式)`

```
> # P < 0.05であるデータを取り出す  
> head(subset(MEA.data, P.value < 0.05))  
> nrow(subset(MEA.data, P.value < 0.05))
```

```
> # P < 0.05であるデータのうち、プロモーター領域にあるCpGを取り出す  
> head(subset(MEA.data, P.value < 0.05 & region == "promoter"), "CpG")
```

ここまでの、プロモーター且つ $P < 0.05$ のデータの、CpG列を取れ！

```
> # exon領域のデータのみを取り出す  
> head(subset(MEA.data, region=="first-exon"|region=="last-exon"|region=="other-exon"))  
> head(subset(MEA.data, region %in% c("first-exon", "last-exon", "other-exon")))
```

A %in% B

ベクトルAの各要素について、
ベクトルBの要素のいずれかが一致すればTRUE、
そうでなければFALSEを返す。

条件式による部分データの抽出2

`data[条件式,]`

```
> # P < 0.05であるデータを取り出す
```

```
> head(MEA.data[MEA.data$P.value<0.001,])
```

```
> # P < 0.005であるデータのうち、プロモーター領域にあるCpGを取り出す
```

```
> head(MEA.data[MEA.data$P.value<0.001&MEA.data$region=="promoter","CpG"])
```

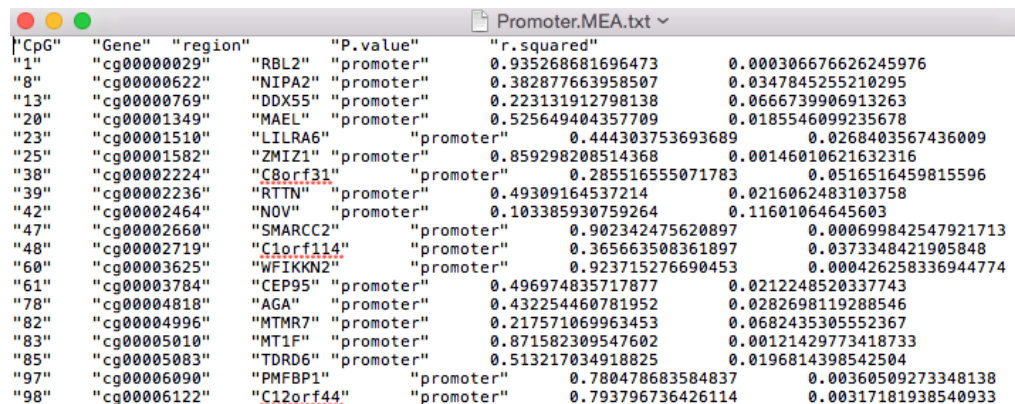
データフレームのファイルへの書き出し

```
write.table(data, file= "", options...)
```

プロモーター領域のみのデータを書き出したい

```
> # プロモーター領域のみのデータを取り出して、Promoter.MEAに代入する
> Promoter.MEA <- subset(MEA.data, region=="promoter")

> # タブ切りテキストとして保存する
> write.table(Promoter.MEA, "Promoter.MEA.txt", sep="\t")
```



CpG	Gene	region	P.value	r.squared	
"1"	"cg00000029"	"RBL2"	"promoter"	0.935268681696473	0.000306676626245976
"8"	"cg000000622"	"NIPA2"	"promoter"	0.382877663958507	0.0347845255210295
"13"	"cg000000769"	"DDX55"	"promoter"	0.223131912798138	0.0666739906913263
"20"	"cg00001349"	"MAEL"	"promoter"	0.525649404357709	0.0185546099235678
"23"	"cg00001510"	"LILRA6"	"promoter"	0.444303753693689	0.0268403567436009
"25"	"cg00001582"	"ZMIZ1"	"promoter"	0.859298208514368	0.00146010621632316
"38"	"cg00002224"	"C8orf31"	"promoter"	0.285516555071783	0.0516516459815596
"39"	"cg00002236"	"RTTN"	"promoter"	0.49309164537214	0.0216062483103758
"42"	"cg00002464"	"NOV"	"promoter"	0.103385930759264	0.11601064645603
"47"	"cg00002660"	"SMARCC2"	"promoter"	0.902342475620897	0.000699842547921713
"48"	"cg00002719"	"C1orf114"	"promoter"	0.365663508361897	0.0373348421905848
"60"	"cg00003625"	"WFIKN2"	"promoter"	0.923715276690453	0.000426258336944774
"61"	"cg00003784"	"CEP95"	"promoter"	0.496974835717877	0.0212248520337743
"78"	"cg00004818"	"AGA"	"promoter"	0.432254460781952	0.0282698119288546
"82"	"cg00004996"	"MTMR7"	"promoter"	0.217571069963453	0.0682435305552367
"83"	"cg00005010"	"MT1F"	"promoter"	0.871582309547602	0.00121429773418733
"85"	"cg00005083"	"TDRD6"	"promoter"	0.513217034918825	0.0196814398542504
"97"	"cg00006090"	"PMFBP1"	"promoter"	0.780478683584837	0.00360509273348138
"98"	"cg00006122"	"C12orf44"	"promoter"	0.793796736426114	0.00317181938540933

```
> # 二重引用符でくくらない
> write.table(Promoter.MEA, "Promoter.MEA.txt", sep="\t", quote=F)

> # 行番号を書き出さない
> write.table(Promoter.MEA, "Promoter.MEA.txt", sep="\t", quote=F, row.names=F)
```

基本統計量を算出してみよう

mammals.txtを読み込む

	row.names	body	brain
1	Arctic fox	3.385	44.50
2	Owl monkey	0.480	15.50
3	Mountain beaver	1.350	8.10
4	Cow	465.000	423.00
5	Grey wolf	36.330	119.50
6	Goat	27.660	115.00
7	Roe deer	14.830	98.20
8	Guinea pig	1.040	5.50
9	Verbet	4.190	58.00
10	Chinchilla	0.425	6.40

body: 体重 (kg)
brain: 脳の重さ(g)

```
> str(mammals)
'data.frame':  62 obs. of  2 variables:
 $ body : num  3.38 0.48 1.35 465 36.33 ...
 $ brain: num  44.5 15.5 8.1 423 119.5 ...
```

【復習】 mammals\$brainの基本統計量

- 欲しい統計量

- ✓ 最小値：min(x)
- ✓ 最大値：max(x)
- ✓ 平均値：mean(x)
- ✓ 中央値：median(x)
- ✓ 標準偏差：sd(x)

```
> min(mammals$brain)
[1] 0.14
> max(mammals$brain)
[1] 5712
> mean(mammals$brain)
[1] 283.1342
> median(mammals$brain)
[1] 17.25
> sd(mammals$brain)
[1] 930.2789
```

- summary関数を使う

```
> summary(mammals$brain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.14   4.25   17.25  283.10  166.00  5712.00
```

mammalsの基本統計量を表にまとめる

- apply関数を使うと便利

```
apply(data, 1 or 2, 関数)
1は行単位, 2は列単位で処理する
```

- 列ごとに統計量を算出して、
mammals.statsに代入する

```
> mammals.stats <- apply(mammals, 2, summary)
> mammals.stats
```

	body	brain
Min.	0.005	0.14
1st Qu.	0.600	4.25
Median	3.342	17.25
Mean	198.800	283.10
3rd Qu.	48.200	166.00
Max.	6654.000	5712.00

```
rbind(data1, data2): 縦に結合
cbind(data1, data2): 横に結合
```

	row.names	body	brain
1	Arctic fox	3.385	44.50
2	Owl monkey	0.480	15.50
3	Mountain beaver	1.350	8.10
4	Cow	465.000	423.00
5	Grey wolf	36.330	119.50
6	Goat	27.660	115.00
7	Roe deer	14.830	98.20
8	Guinea pig	1.040	5.50
9	Verbet	4.190	58.00
10	Chinchilla	0.425	6.40

- 標準偏差を算出し、
mammals.statsに加える

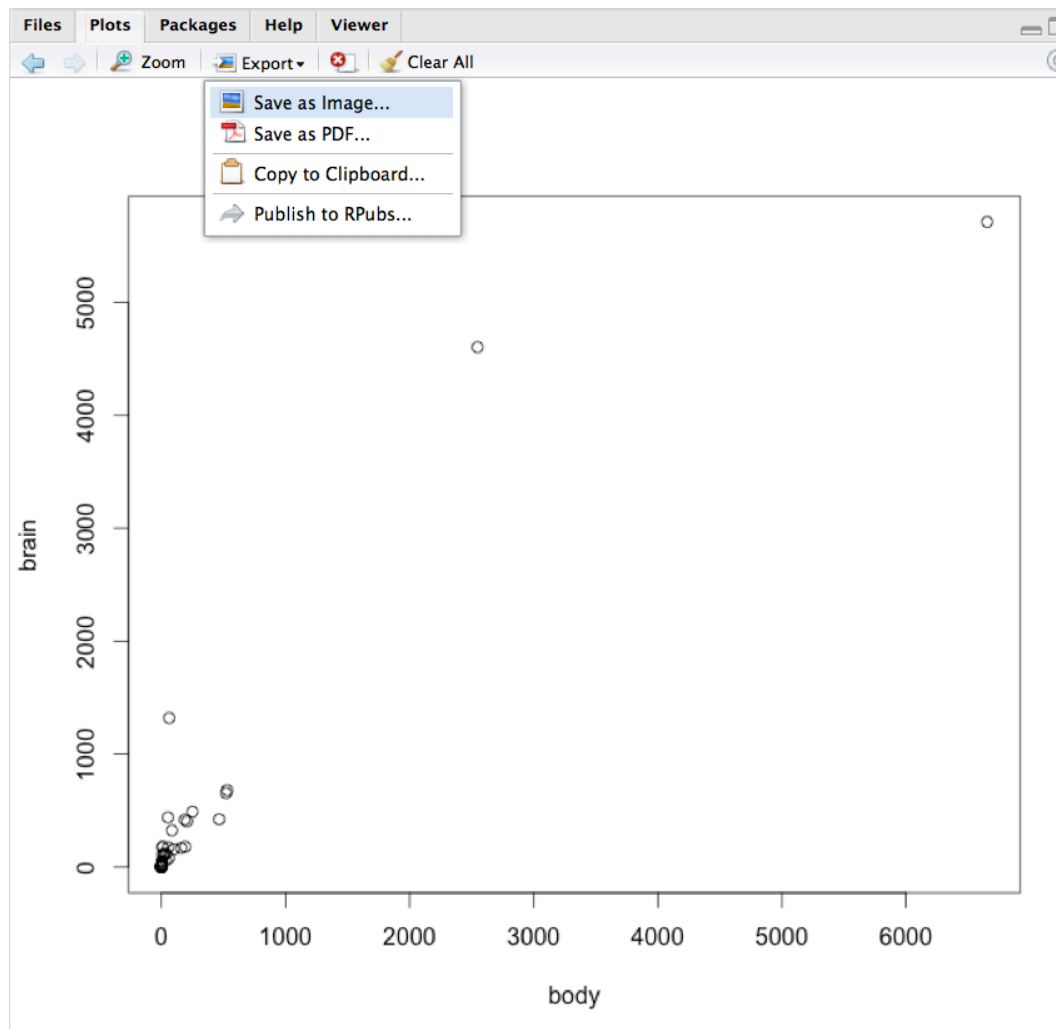
```
> SD <- apply(mammals, 2, sd)
> mammals.stats <- rbind(mammals.stats, SD)
> mammals.stats
```

	body	brain
Min.	0.005	0.1400
1st Qu.	0.600	4.2500
Median	3.342	17.2500
Mean	198.800	283.1000
3rd Qu.	48.200	166.0000
Max.	6654.000	5712.0000
SD	899.158	930.2789

プロットを描く

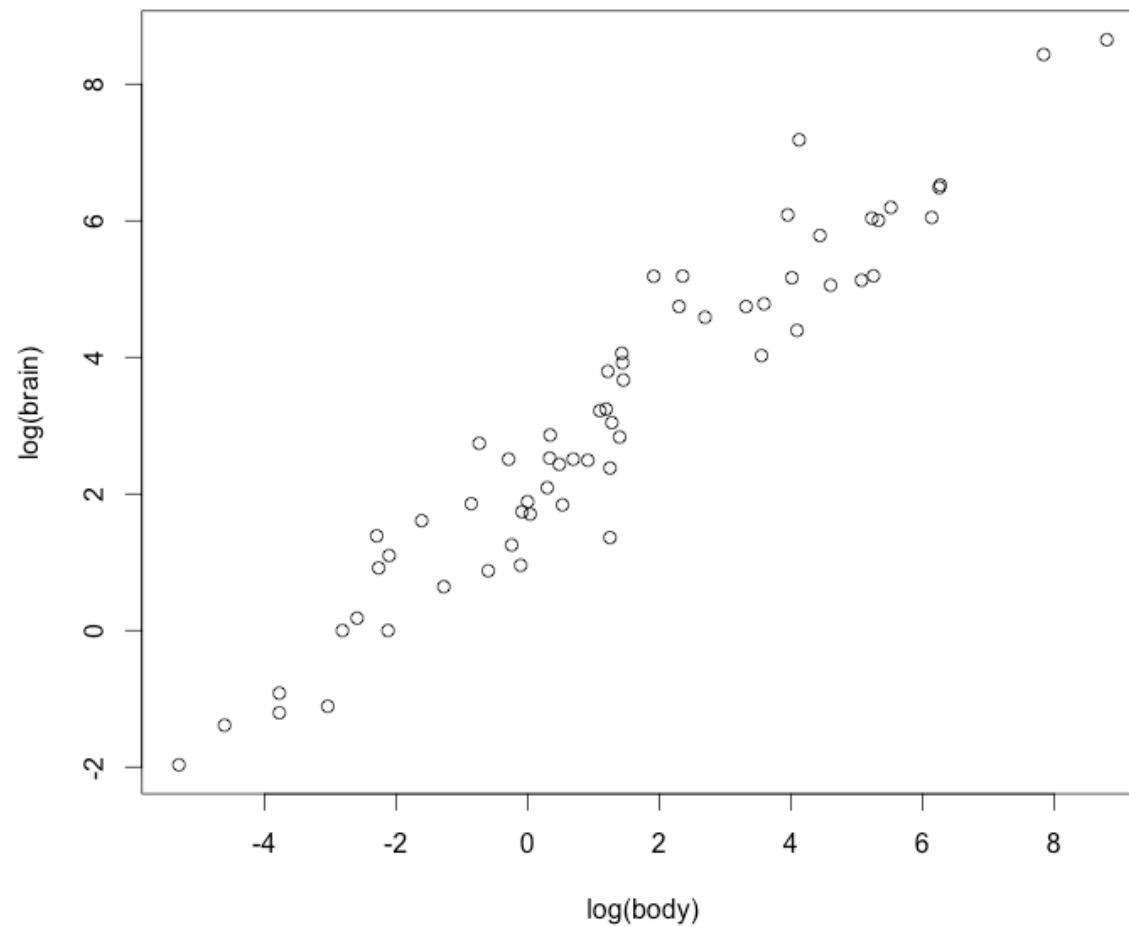
```
> plot(mammals$body,mammals$brain)
```

```
> plot(brain~body,data=mammals) ← おすすめ
```



logスケールでプロットを描く

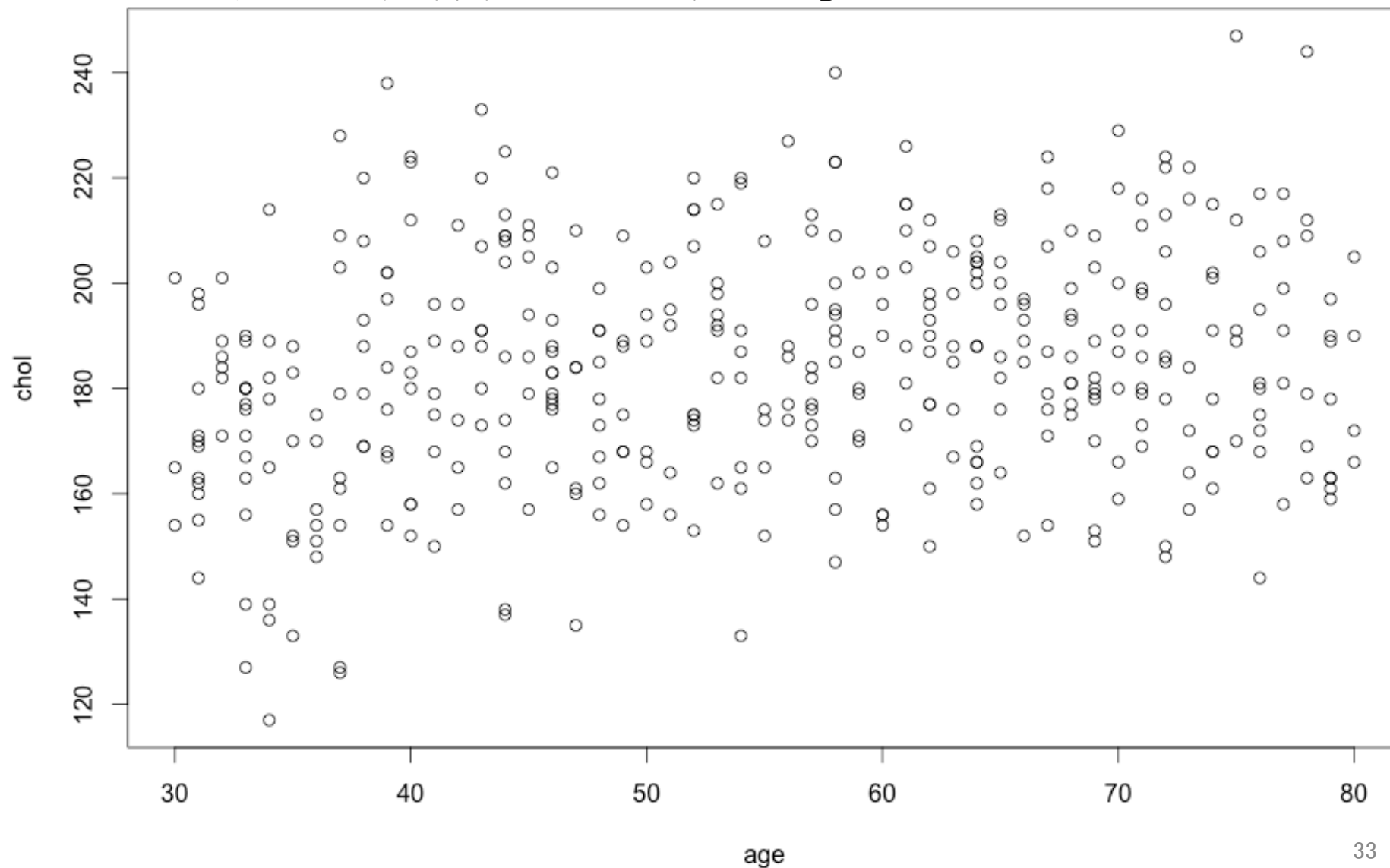
```
> plot(log(brain)~log(body),data=mammals)
```



回帰分析をやってみよう

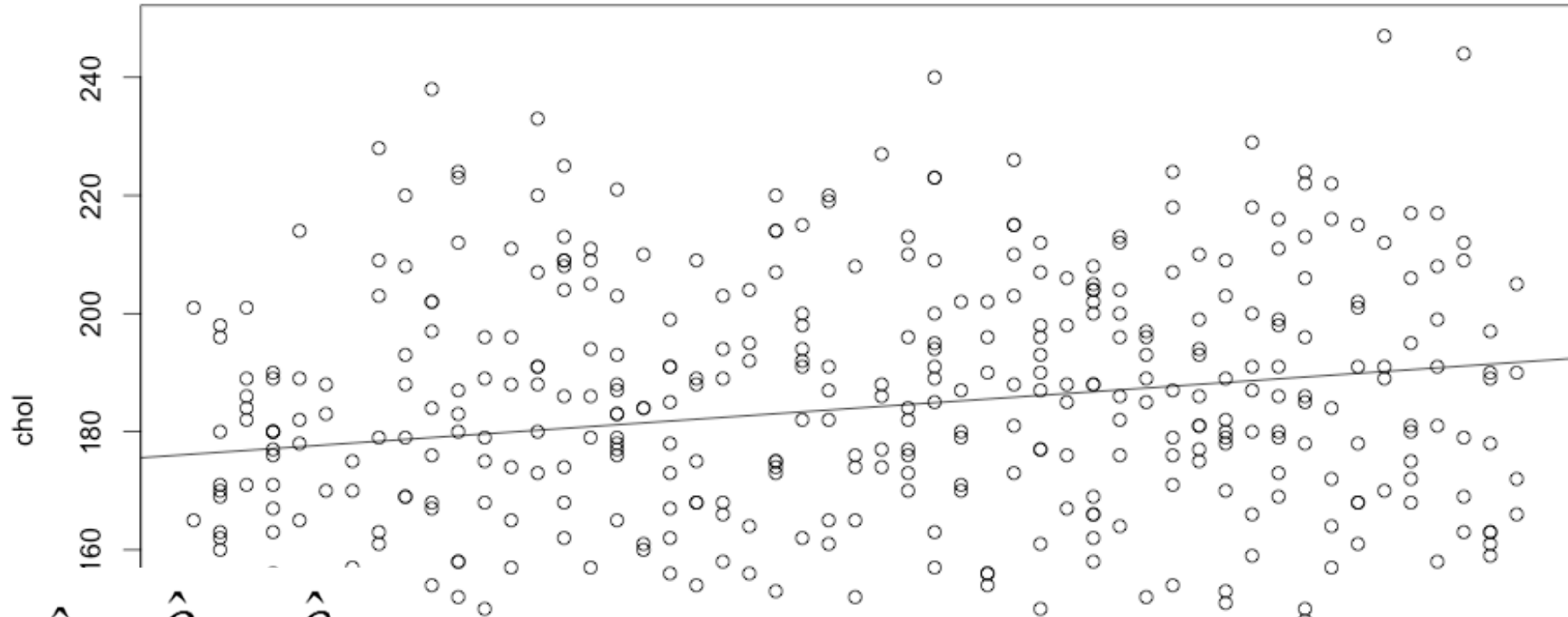
コレステロール値と年齢の関係

- 2つの変数の関係性を「客観的に」記述するには・・・？



コレステロール値と年齢の関係

- 直線（回帰直線）を描く → 傾きと切片 → 傾きの有意性



$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

$$\hat{\beta}_0 = 166.90 \quad 95\% \text{ CI: } (158.5, 175.3) \quad P < 2 \times 10^{-16}$$

$$\hat{\beta}_1 = 0.31 \quad 95\% \text{ CI: } (0.16, 0.46) \quad P = 4.52 \times 10^{-5}$$

mammalsデータの線形回帰 (log-scale)

```
> mammals.reg.log = lm( log(brain) ~ log(body), data = mammals )
> plot( log(brain) ~ log(body), data = mammals )
> par(new=T)
> abline(mammals.reg.log)
> summary(mammals.reg.log)
```

Call:

```
lm(formula = log(brain) ~ log(body), data = mammals)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.71550	-0.49228	-0.06162	0.43597	1.94829

Coefficients:

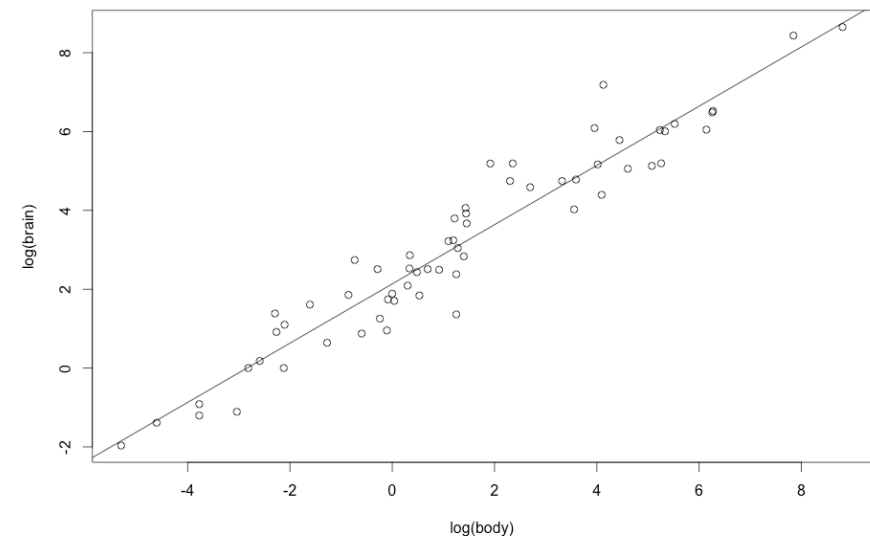
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.13479	0.09604	22.23	<2e-16 ***
log(body)	0.75169	0.02846	26.41	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6943 on 60 degrees of freedom

Multiple R-squared: 0.9208, Adjusted R-squared: 0.9195

F-statistic: 697.4 on 1 and 60 DF, p-value: < 2.2e-16



$$\log(\text{Brain}) = 2.13 + 0.75 \times \log(\text{Body})$$

傾きと切片、有意性、相関係数

```
> summary(mammals.reg.log)
```

Call:

```
lm(formula = log(brain) ~ log(body), data = mammals)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.71550	-0.49228	-0.06162	0.43597	1.94829

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.13479	0.09604	22.23	<2e-16	***
log(body)	0.75169	0.02846	26.41	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6943 on 60 degrees of freedom

Multiple R-squared: 0.9208, Adjusted R-squared: 0.9195

F-statistic: 697.4 on 1 and 60 DF, p-value: < 2.2e-16

$$R^2 = 0.92$$

```
> confint(mammals.reg.log)
```

	2.5 %	97.5 %
(Intercept)	1.9426733	2.3269041
log(body)	0.6947503	0.8086215

$$\hat{\beta}_0 = 2.13 \ [1.94 - 2.33] \quad P < 2 \times 10^{-16}$$

$$\hat{\beta}_1 = 0.75 \ [0.69 - 0.81] \quad P < 2 \times 10^{-16}$$

回帰分析の結果から情報を取り出す

```
> names(summary(mammals.reg.log))
```

```
[1] "call"          "terms"          "residuals"      "coefficients"  
[5] "aliased"        "sigma"          "df"             "r.squared"  
[9] "adj.r.squared" "fstatistic"     "cov.unscaled"
```

```
> summary(mammals.reg.log)$coefficients
```

```
              Estimate Std. Error  t value    Pr(>|t|)  
(Intercept)  2.1347887  0.09604339  22.22734 1.183207e-30  
log(body)     0.7516859  0.02846356  26.40871 9.835792e-35
```

2行4列のデータ！

```
> # 傾き
```

```
> summary(mammals.reg.log)$coefficients[2,1]
```

```
[1] 0.7516859
```

```
> # 傾きの有意性 (P値)
```

```
> summary(mammals.reg.log)$coefficients[2,4]
```

```
[1] 9.835792e-35
```

```
> # 寄与率
```

```
> summary(mammals.reg.log)$r.squared
```

```
[1] 0.9207837
```

```
> # 傾きの95%CI
```

```
> confint(mammals.reg.log)[2,]
```

```
      2.5 %      97.5 %  
0.6947503 0.8086215
```

カテゴリーデータの取り扱い ～分散分析（ANOVA）～

lung_cancer.txtを読み込む

	ID_REF	tissue	smoking	stage	gender	gene1	gene2	gene3	gene4	gene5	gene6
1	GSM254629	tumor	never	stage I	female	7.41910	5.93180	5.67496	6.06873	7.26279	5.02459
2	GSM254648	tumor	never	stage I	female	7.56270	6.93398	5.76701	8.24300	8.13711	4.94092
3	GSM254694	tumor	never	stage I	female	7.54599	7.53287	5.84134	7.13335	8.26834	5.11204
4	GSM254701	tumor	never	stage I	female	8.31452	7.88291	5.44759	5.99769	7.66485	5.06010
5	GSM254728	tumor	never	stage I	female	7.19835	6.58398	4.79089	7.26575	7.46492	5.02376
6	GSM254726	tumor	never	stage I	male	11.98110	8.45595	5.70830	8.30360	8.36494	8.69256

- 肺がん患者の遺伝子発現プロフィールデータ（GEO: GDS3257）の抜粋

```
> str(lung_cancer)
```

```
'data.frame': 107 obs. of 11 variables:
```

```
$ ID_REF : Factor w/ 107 levels "GSM254625","GSM254626",...: 5 24 70 77 104 102 15 28 76 1 ...
```

```
$ tissue : Factor w/ 2 levels "normal","tumor": 2 2 2 2 2 2 2 2 2 2 ...
```

```
$ smoking: Factor w/ 3 levels "current","former",...: 3 3 3 3 3 3 3 3 3 3 ...
```

```
$ stage : Factor w/ 4 levels "stage I","stage II",...: 1 1 1 1 1 1 2 2 2 2 ...
```

```
$ gender : Factor w/ 2 levels "female","male": 1 1 1 1 1 2 1 1 1 2 ...
```

```
$ gene1 : num 7.42 7.56 7.55 8.31 7.2 ...
```

```
$ gene2 : num 5.93 6.93 7.53 7.88 6.58 ...
```

```
$ gene3 : num 5.67 5.77 5.84 5.45 4.79 ...
```

```
$ gene4 : num 6.07 8.24 7.13 6 7.27 ...
```

```
$ gene5 : num 7.26 8.14 8.27 7.66 7.46 ...
```

```
$ gene6 : num 5.02 4.94 5.11 5.06 5.02 ...
```

tissue: 腫瘍 or 正常

smoking: 喫煙歴

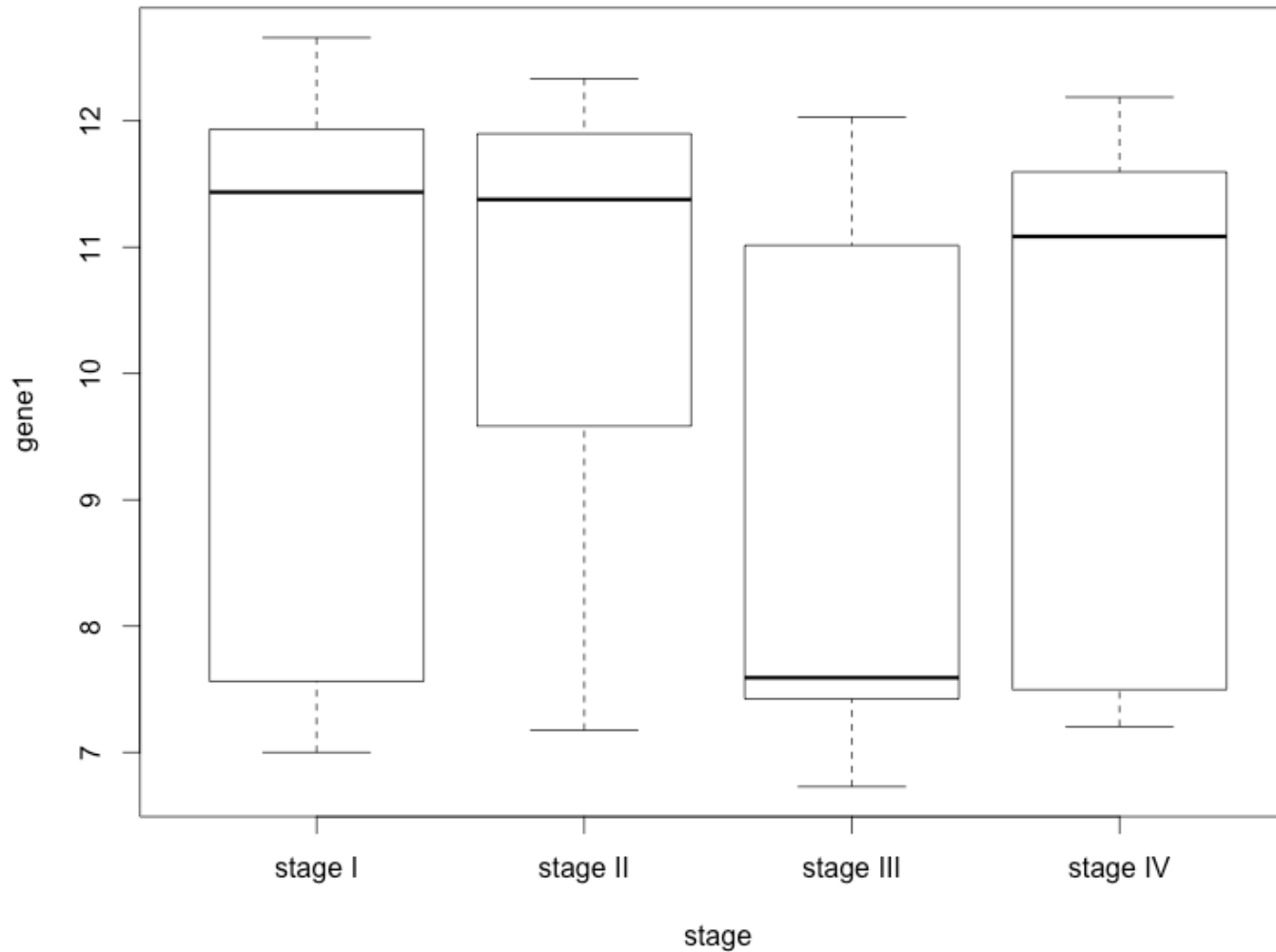
stage: 癌のステージ

gender: 性別

gene1-gene6: 遺伝子発現データ

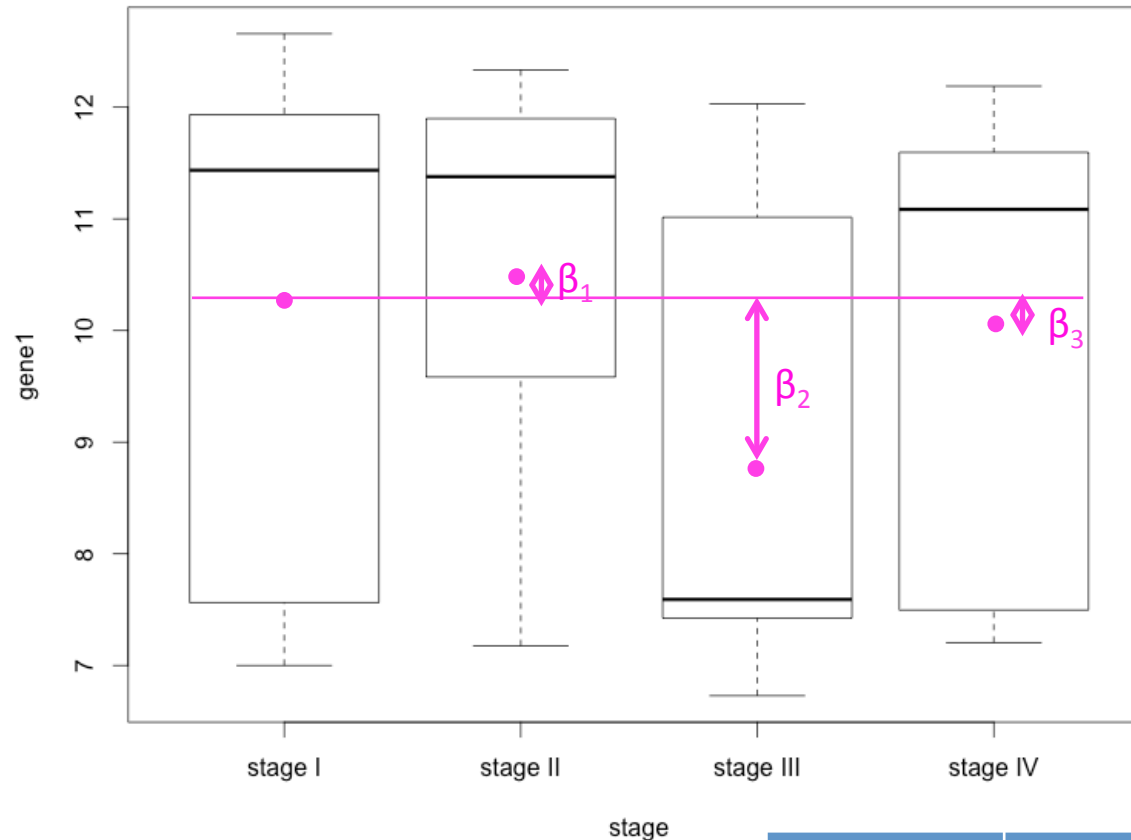
gene1の発現量とstageの関係

```
> plot(gene1~stage,data=lung_cancer)
```



gene1の発現と肺がんのステージに関係性はあるか？

gene1の発現量とstageの関係を式で書く



$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$$

β_0 : stage Iの平均値

$\beta_0 + \beta_1$: stage IIの平均値

$\beta_0 + \beta_2$: stage IIIの平均値

$\beta_0 + \beta_3$: stage IVの平均値

stage	x_1	x_2	x_3
stage I	0	0	0
stage II	1	0	0
stage III	0	1	0
stage IV	0	0	1

Rで計算

```
> gene1.stage <- lm(gene1~stage,data=lung_cancer)
> summary(gene1.stage)
```

Call:

```
lm(formula = gene1 ~ stage, data = lung_cancer)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-3.3522 -1.5056  0.9377  1.4800  3.2804
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.3335	0.2936	35.191	< 2e-16 ***
stagestage II	0.1945	0.4439	0.438	0.66216
stagestage III	-1.5835	0.5206	-3.042	0.00298 **
stagestage IV	-0.2244	0.8561	-0.262	0.79380

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.97 on 103 degrees of freedom

Multiple R-squared: 0.1054, Adjusted R-squared: 0.0793

F-statistic: 4.043 on 3 and 103 DF, p-value: 0.009211

```
> confint(gene1.stage)
```

	2.5 %	97.5 %
(Intercept)	9.7511576	10.9158975
stagestage II	-0.6859248	1.0749965
stagestage III	-2.6159696	-0.5511064
stagestage IV	-1.9222365	1.4735348

stage	X ₁	X ₂	X ₃	Yの平均
stage I	0	0	0	10.3
stage II	1	0	0	10.3+0.2
stage III	0	1	0	10.3-1.6
stage IV	0	0	1	10.3-0.2

$$\beta_0 = 10.3 \text{ (95\% CI: } 9.8 - 10.9; P < 2.0 \times 10^{-16})$$

$$\beta_1 = 0.2 \text{ (95\% CI: } -0.7 - 1.1; P = 0.66)$$

$$\beta_2 = -1.6 \text{ (95\% CI: } -2.6 - -0.6; P = 0.003)$$

$$\beta_3 = -0.2 \text{ (95\% CI: } -1.9 - 1.5; P = 0.79)$$

カテゴリー変数のP値

```
> gene1.stage <- lm(gene1~stage,data=lung_cancer)
> summary(gene1.stage)
```

Call:

```
lm(formula = gene1 ~ stage, data = lung_cancer)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.3522	-1.5056	0.9377	1.4800	3.2804

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.3335	0.2936	35.191	< 2e-16 ***
stagestage II	0.1945	0.4439	0.438	0.66216
stagestage III	-1.5835	0.5206	-3.042	0.00298 **
stagestage IV	-0.2244	0.8561	-0.262	0.79380

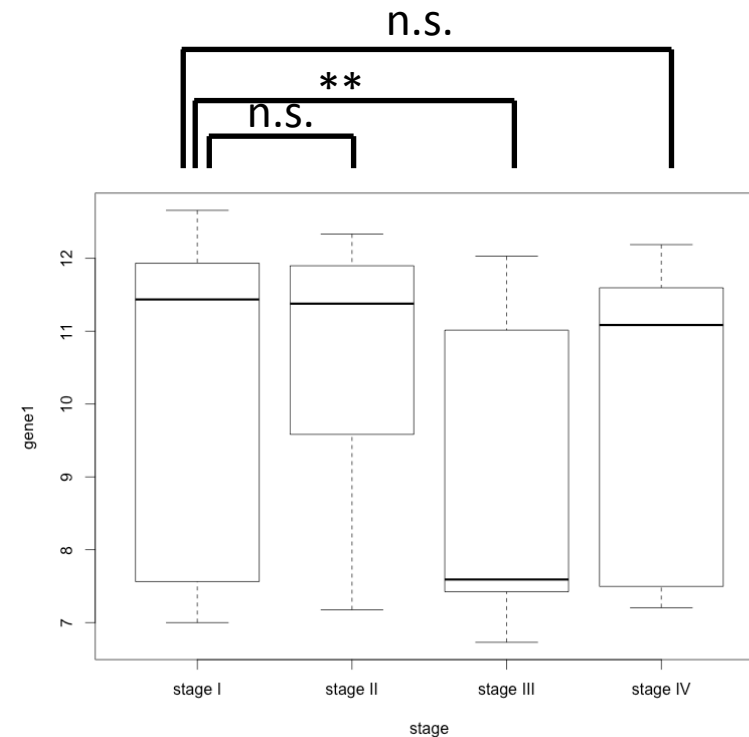
```
> anova(gene1.stage)
```

Analysis of Variance Table

Response: gene1

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
stage	3	47.07	15.6892	4.0434	0.009211 **
Residuals	103	399.66	3.8802		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0$$

$$H_1 : \text{not all means are zero}$$

$$P = 0.009211$$

すべての遺伝子でstageとの関係性を分析する

		β_1	β_1 のP値	β_2	β_2 のP値	β_3	β_3 のP値	stageのP値
	row.names	coe_stage2	Pval_stage2	coe_stage3	Pval_stage3	coe_stage4	Pval_stage4	Pval_stage.cat
1	gene1	0.1945359	0.6621605	-1.58353803	0.002982150	-0.2243509	0.7937989	0.009211405
2	gene2	-0.1035874	0.8340672	-0.27341051	0.637402945	0.2417104	0.7999044	0.948740645
3	gene3	-0.4938128	0.2211256	-0.18931225	0.688178817	-0.1799234	0.8165436	0.678361617
4	gene4	-0.4791879	0.2089383	-0.29791025	0.504122268	-0.2388098	0.7445096	0.650429117
5	gene5	-0.4287010	0.1724967	-0.08907089	0.808169986	-0.3799459	0.5291994	0.553112938
6	gene6	0.2077314	0.6074477	-1.57633283	0.001187868	-0.3104621	0.6904524	0.003151700

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$$

```
> summary(gene1.stage)$coefficients
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.3335276	0.2936422	35.1908849	3.186869e-59
stagestage II	0.1945359	0.4439452	0.4381979	6.621605e-01
stagestage III	-1.5835380	0.5205719	-3.0419198	2.982150e-03
stagestage IV	-0.2243509	0.8561067	-0.2620595	7.937989e-01

4行4列のデータ！

```
> anova(gene1.stage)
```

Analysis of Variance Table

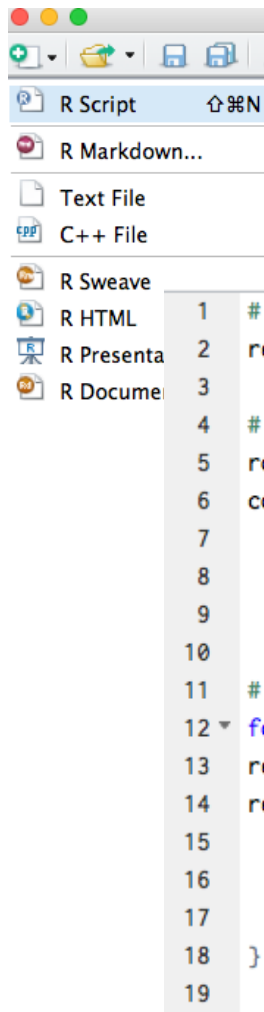
2行5列のデータ！

Response: gene1					
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
stage	3	47.07	15.6892	4.0434	0.009211 **
Residuals	103	399.66	3.8802		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

同じ処理を繰り返す (for文)

```
for(ループ変数 in ベクトル){  
  繰り返したい処理  
}
```



	ID_REF	tissue	smoking	stage	gender	gene1	gene2	gene3	gene4	gene5	gene6
1	GSM254629	tumor	never	stage I	female	7.41910	5.93180	5.67496	6.06873	7.26279	5.02459
2	GSM254648	tumor	never	stage I	female	7.56270	6.93398	5.76701	8.24300	8.13711	4.94092
3	GSM254694	tumor	never	stage I	female	7.54599	7.53287	5.84134	7.13335	8.26834	5.11204
4	GSM254701	tumor	never	stage I	female	8.31452	7.88291	5.44759	5.99769	7.66485	5.06010
5	GSM254728	tumor	never	stage I	female	7.19835	6.58398	4.79089	7.26575	7.46492	5.02376
6	GSM254726	tumor	never	stage I	male	11.98110	8.45595	5.70830	8.30360	8.36494	8.69256

回帰分析の結果をresに格納する (6列目がgene1)



多重性の問題とP値の補正

- 各遺伝子の検定で、 $P < 0.05$ で棄却したリストを集めた場合、実際に得られたリストは想定以上の偽陽性を含んでいる。

例) $P < 0.05$ で2つの遺伝子を検定した場合の偽陽性を含む確率

$$1 - (1 - 0.05)^2 = 0.0975$$

- 最終的に得られる遺伝子リストが目的の閾値に収まるように、個々の遺伝子のP値を補正する必要がある。
 - ボンフェローニの補正：閾値を繰り返す検定数で割る
1000個の遺伝子を検定した場合、閾値の0.05を1000で割る
メリット：必ず設定した閾値を担保できる
デメリット：数千～数万回の検定の場合、
非常に小さなP値のみしか閾値を満たさない
 - BenjaminiらのFalse Discovery Rate (FDR)
得られた遺伝子リストの中の偽陽性率をコントロールできる

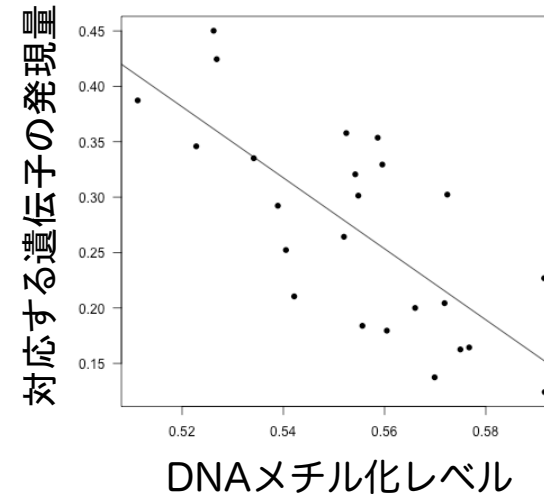
P値を補正して有意差があるリストを得る

```
> head(MEA.data)
```

	CpG	Gene	region	P.value	r.squared
1	cg00000029	RBL2	promoter	0.9352687	0.0003066766
2	cg00000108	C3orf35	last-exon	0.0537268	0.1588396212
3	cg00000109	FNDC3B	intron	0.3100466	0.0467821335
4	cg00000236	VDAC3	last-exon	0.1276800	0.1022628986
5	cg00000289	ACTN1	last-exon	0.8820903	0.0010223983
6	cg00000292	ATP2A1	first-exon	0.8216525	0.0023600131

```
> dim(MEA.data)
```

```
[1] 276663      5
```



- 同一個人から3ヶ月で24回採血し、セルソーターで、特定の細胞種を分取
- マイクロアレイ (Illumina HM450) で約48万ヶ所のCpGサイトのDNAメチル化データを得た
- RNA-seqにより、約2万遺伝子の発現量データを得た
- 各CpGサイトにおけるDNAメチル化レベルの変動が、近傍の遺伝子の発現変動に寄与しているか、回帰分析を行った

```
lm(gene expression ~ DNAm)
```


ボンフェローニによる補正

```
> 0.05/nrow(MEA.data)
[1] 1.807253e-07
> subset(MEA.data,P.value < 0.05/nrow(MEA.data))
```

	CpG	Gene	region	P.value	r.squared
98889	cg09032544	CD247	intron	5.051998e-08	0.7477331
191604	cg18417464	LUC7L2	first-exon	1.034078e-07	0.7310065
216485	cg21161394	CD247	intron	1.034926e-07	0.7309868

FDRによる補正

```
> MEA.data[,6] <- p.adjust(MEA.data$P.value,method="fdr")
> head(MEA.data)
```

	CpG	Gene	region	P.value	r.squared	V6
1	cg00000029	RBL2	promoter	0.9352687	0.0003066766	0.9999655
2	cg00000108	C3orf35	last-exon	0.0537268	0.1588396212	0.9999655
3	cg00000109	FNDC3B	intron	0.3100466	0.0467821335	0.9999655
4	cg00000236	VDAC3	last-exon	0.1276800	0.1022628986	0.9999655
5	cg00000289	ACTN1	last-exon	0.8820903	0.0010223983	0.9999655
6	cg00000292	ATP2A1	first-exon	0.8216525	0.0023600131	0.9999655

```
> subset(MEA.data,V6 < 0.05)
```

	CpG	Gene	region	P.value	r.squared	V6
92492	cg08402433	CARD11	intron	2.659872e-07	0.7072740	0.018397205
98889	cg09032544	CD247	intron	5.051998e-08	0.7477331	0.009544195
191604	cg18417464	LUC7L2	first-exon	1.034078e-07	0.7310065	0.009544195
216485	cg21161394	CD247	intron	1.034926e-07	0.7309868	0.009544195

本日のまとめ

- Rの基本操作
- 統計量を算出して表にまとめる
- 回帰分析／分散分析を行い、分析結果から情報を取り出してまとめる
- 多重性の問題を補正したP値で回避する

皆さんの今後の解析の参考になれば幸いです。