第一回　基礎ゼミ
Data aggregation and Visualization

2018.4.25
Fukuda Lab.

Natsuho Ihoroi

Outline

- Introduction ~before starting “R”
- Basic calculation
- Data Visualization
- If文 and for文
- Package

Advantages of programming

It is possible

- to analysis enomous amount of data
- to carry out troublesome processing efficiently

What is R

Statistical analysis software for statiscal processing

Merit

The library is substantial

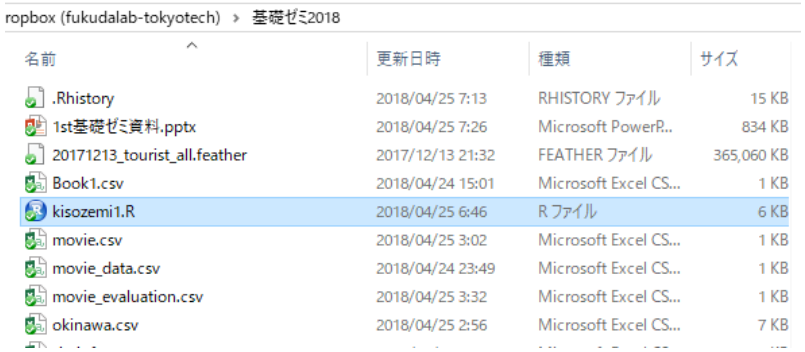
Demerit

Processing of
“for statement” is slow

How to open R files

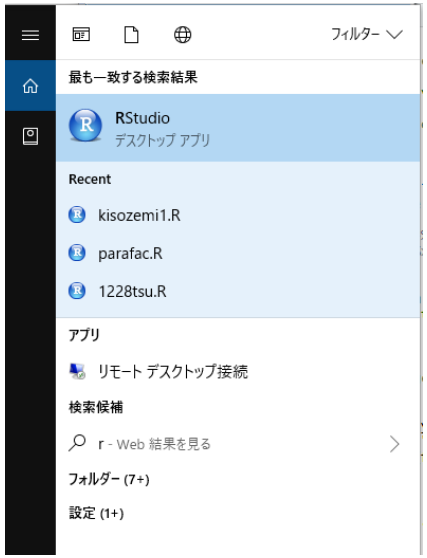
Method 1

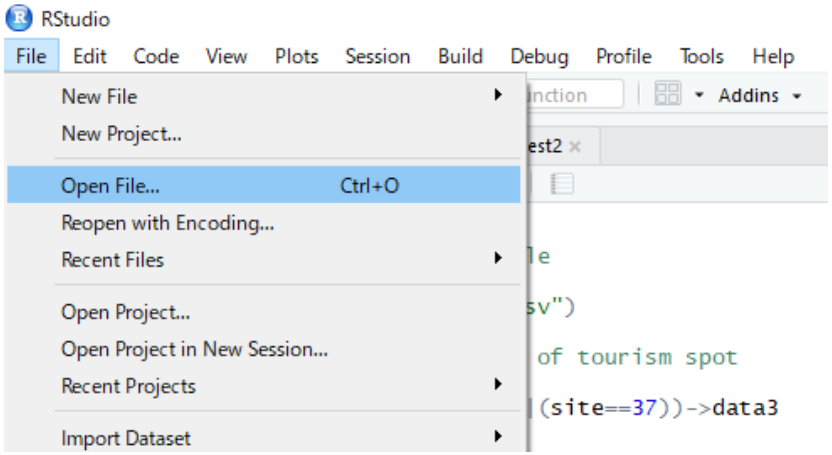
Click the file directly



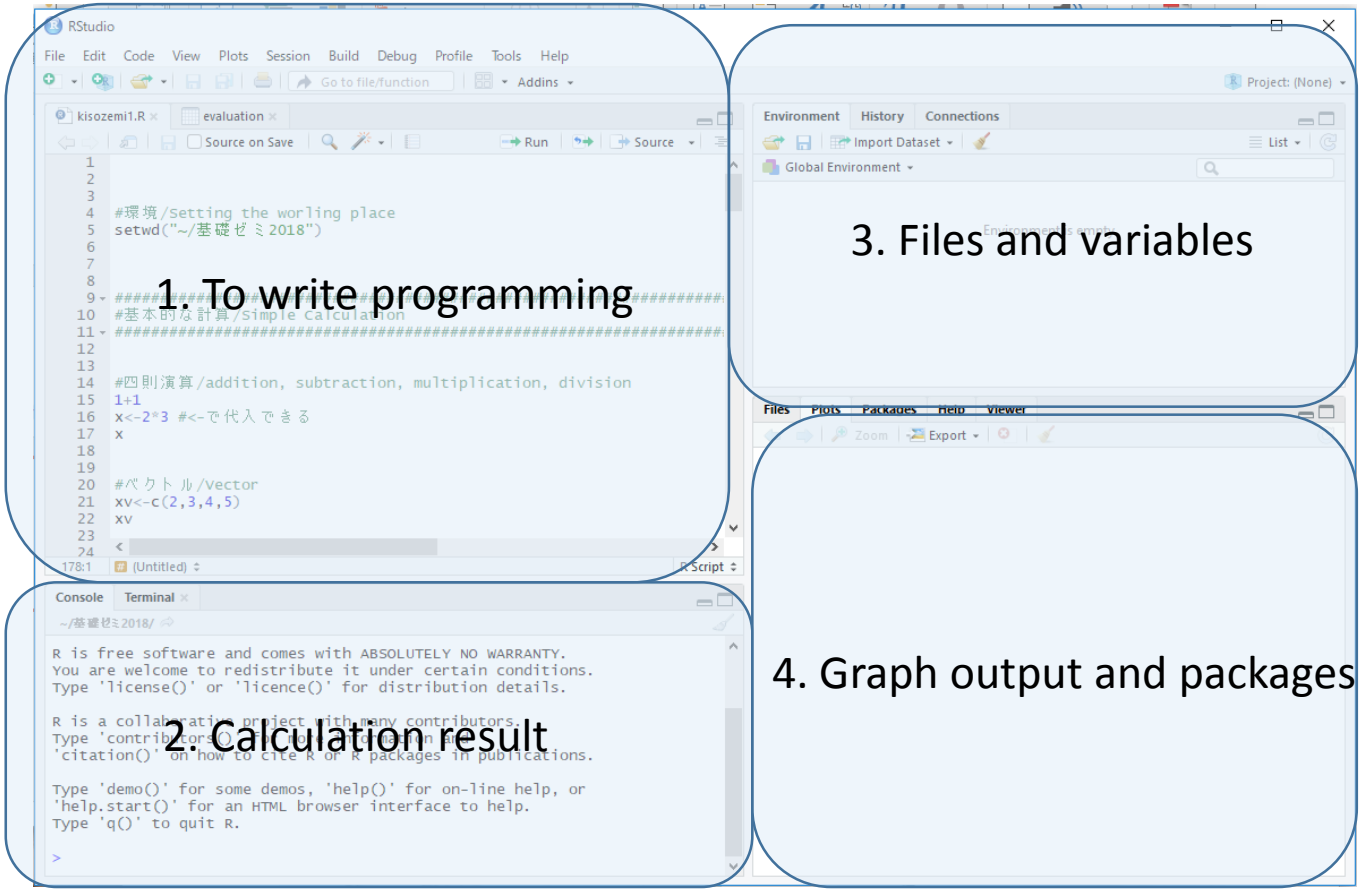
Method 2

Serch “R” on the home screen, and open file from the upper left button





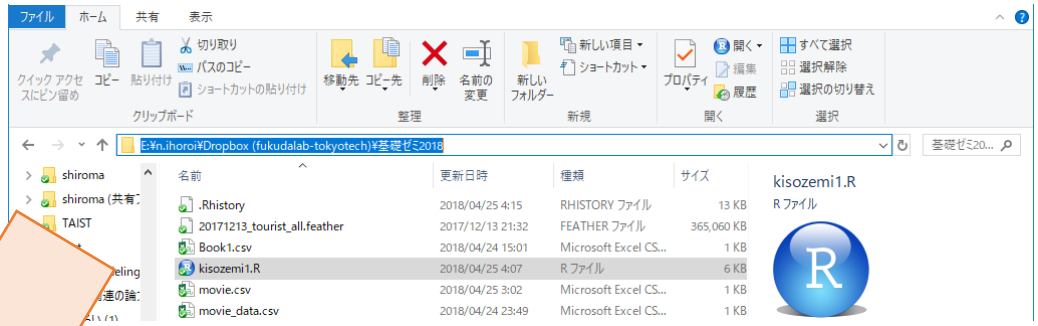
The R screen


1. To write programming
2. Calculation result
3. Files and variables
4. Graph output and packages

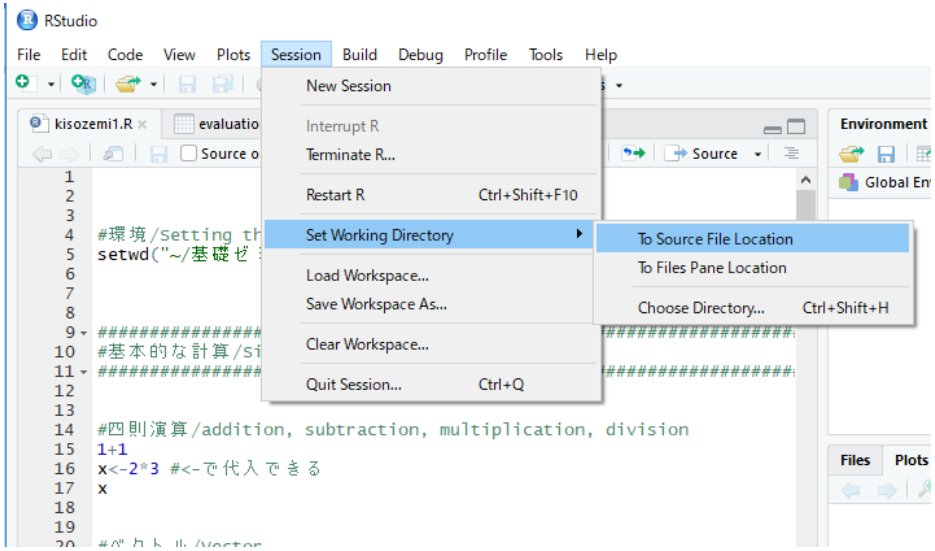
Setting the Working Space

Method 1

Copy and type by yourself
(change: “＼” → “/”)




```
6
7  setwd(E:/n.ihoroi/Dropbox (fukudalab-tokyotech)/基礎ゼミ2018)
8
```



Method 2

Session
→set working directory
→to source file location

How to execute

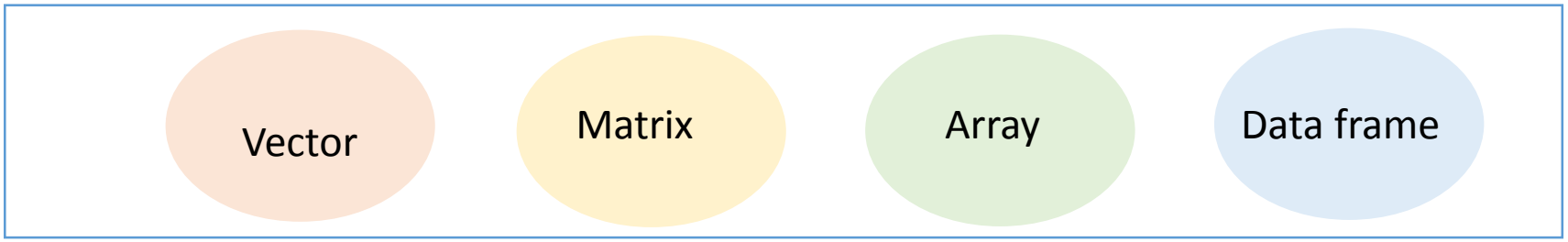
1. write the script/code
2. Select the range you want to execute
3. Ctrl + Enter

Simple Calculation

```
10
11
12  #四則演算/addition, subtraction, multiplication, division
13  1+1
14  x<-2*3 #<-で代入できる
15  x
16
```

Addition	+
Subtraction	-
Multiplication	*
Division	/

Data structure



Vector　1次元，長さを持つベクトル

```
17
18  #ベクトル/Vector
19  xv<-c(2,3,4,5)
20  xv
21
```

C(, , ,)

Matrix　2次元配列

```
22
23  #行列/Matrix
24
25  #作り方1
26  a<-c(1,2,3,4)
27  b<-c(5,6,7,8)
28  xm1<-rbind(a,b)  #row bind(行の結合)
29  xm1
30
31  #作り方2
32  xm2<-matrix(1:8,nrow=2,ncol=4)
33  xm2
34  xm2[1,2]  #1行2列目の値
35  xm2[1,]   #1行目の値
36
```

rbind(a, b) : Row bind
cbind(a, b) : Column bind
Matrix A　Matrix B

Matrix(1:8, nrow=2, ncol=4)
data　Number of rows　Number of colmins

Access to matrix data

X[1, 2]
row　column

Data structure

Array　　行列の多次元拡張

```
37
38  #配列/Array
39  xa<-array(1:24,dim=c(2,4,3))   #2x4x3の3次元配列
40  xa[2,3,]
41  xa[,,1]
42
```

Array(1:24, dim=c(2,4,3)
dimension

Useful function for vector, matrix and array

apply function

```
43
44  #apply関数/apply function
45  apply(xa,c(1,2),mean)   #行列や配列に使える関数！
46
47
```

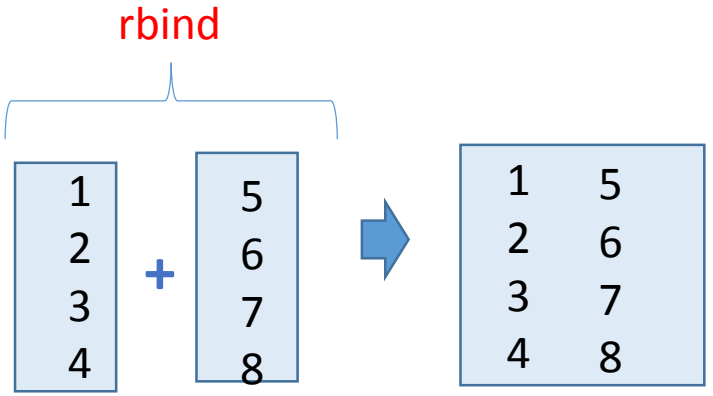
Apply(x, 1, mean)

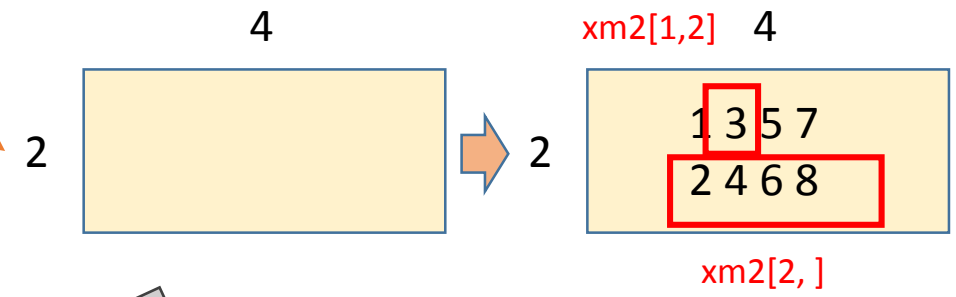
Data　1:row　Function
2:column　(mean, sum,,,)
C(1,2):element

Data structure

Matrix

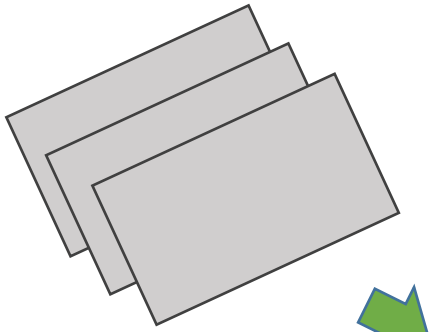
```
22
23  #行列/Matrix
24
25  #作り方1
26  a<-c(1,2,3,4)
27  b<-c(5,6,7,8)
28  xm1<-rbind(a,b)  #row bind(行の結合)
29  xm1
30
31  #作り方2
32  xm2<-matrix(1:8,nrow=2,ncol=4)
33  xm2
34  xm2[1,2]  #1行2列目の値
35  xm2[1,]   #1行目の値
36
```

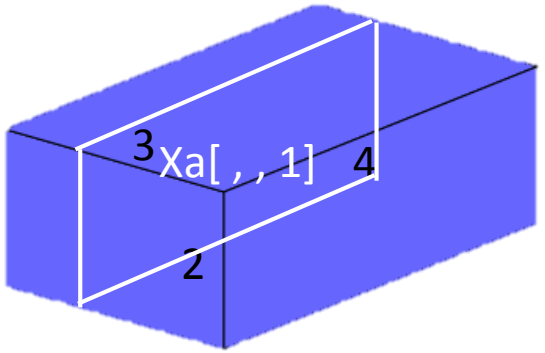




Array

```
37
38  #配列/Array
39  xa<-array(1:24,dim=c(2,4,3))  #2x4x3の3次元配列
40  xa[2,3,]
41  xa[,,1]
42
```





Read the csv file

Data frame

The data is basically “csv file”

“xlsx file” cannot be read

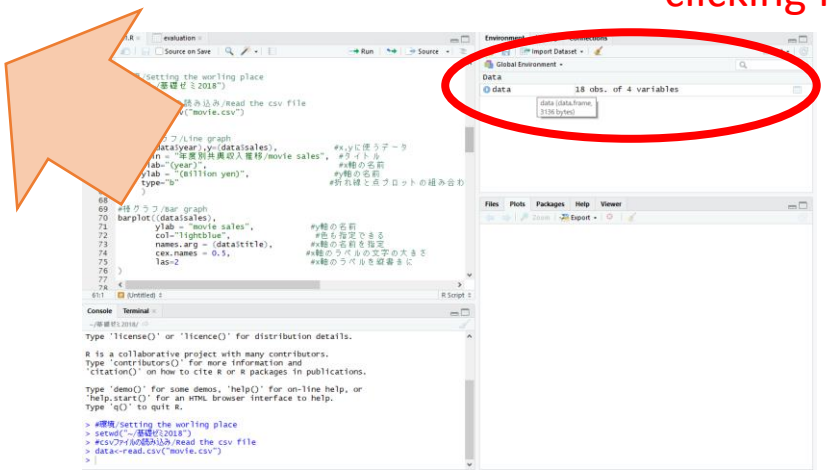
```
56
57  #csvファイルの読み込み/Read the csv file
58  data<-read.csv("movie.csv")
59
```

Read.csv(“file name”)

	title	year	sales	audience
2	風の谷のナウシカ	1984	14.8	91
3	天空の城ラピュタ	1986	11.6	77
4	となりのトトロ	1988	11.7	80
5	魔女の宅急便	1989	36.5	264
6	おもひでぽろぽろ	1991	31.8	216
7	紅の豚	1992	47.6	304
8	平成狸合戦ぽんぽこ	1994	44.7	325
9	耳をすませば	1995	31.5	208
10	もののけ姫	1997	193	1420
11	ホーホケキョ となりの山田くん	1999	15.6	115
12	千と千尋の神隠し	2001	304	2350
13	猫の恩返し	2002	64.6	550
14	ハウルの動く城	2004	196	1500
15	ゲド戦記	2006	76.5	588
16	崖の上のポニョ	2008	155	1200
17	借りぐらしのアリエッティ	2010	92.5	750
18	コクリコ坂から	2011	44.6	355
19	風立ちぬ	2013	120	1000

Movie sales data

You can check the data by clicking here



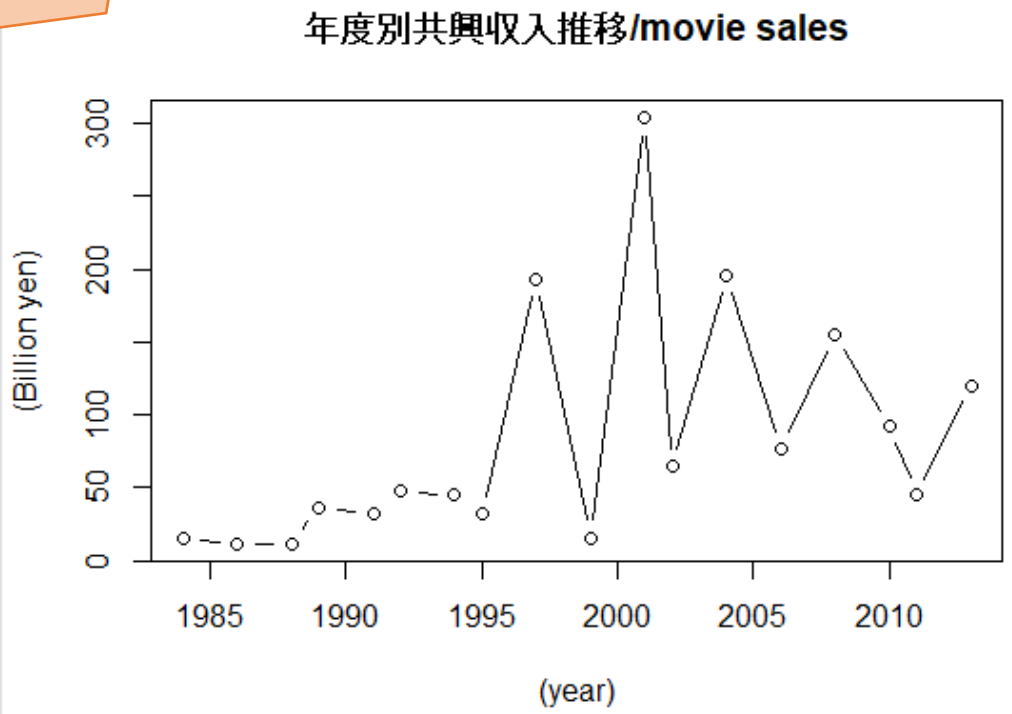
Line graph

```
60
61  #折れ線グラフ/Line graph
62  plot(x=(data$year),y=(data$sales),              #x,yに使うデータ
63       main = "年度別共興収入推移/movie sales",  #タイトル
64       xlab="(year)",                            #x軸の名前
65       ylab = "(Billion yen)",                   #y軸の名前
66       type="b"                                  #折れ線と点プロットの組み合わ
67       )
```

plot(data)

	title	year	ales	audience
2	風の谷のナウシカ	1984	14.8	91
3	天空の城ラピュタ	1986	11.6	77
4	となりのトトロ	1988	11.7	80
5	魔女の宅急便	1989	36.5	264
6	おもひでぽろぽろ	1991	31.8	216
7	紅の豚	1992	47.6	304
8	平成狸合戦ぽんぽこ	1994	44.7	325
9	耳をすませば	1995	31.5	208
10	もののけ姫	1997	193	1420
11	ホーホケキョ となりの山田くん	1999	15.6	115
12	千と千尋の神隠し	2001	304	2350
13	猫の恩返し	2002	64.6	550
14	ハウルの動く城	2004	196	1500
15	ゲド戦記	2006	76.5	588
16	崖の上のポニョ	2008	155	1200
17	借りぐらしのアリエッティ	2010	92.5	750
18	コクリコ坂から	2011	44.6	355
19	風立ちぬ	2013	120	1000

X axis　Y axis



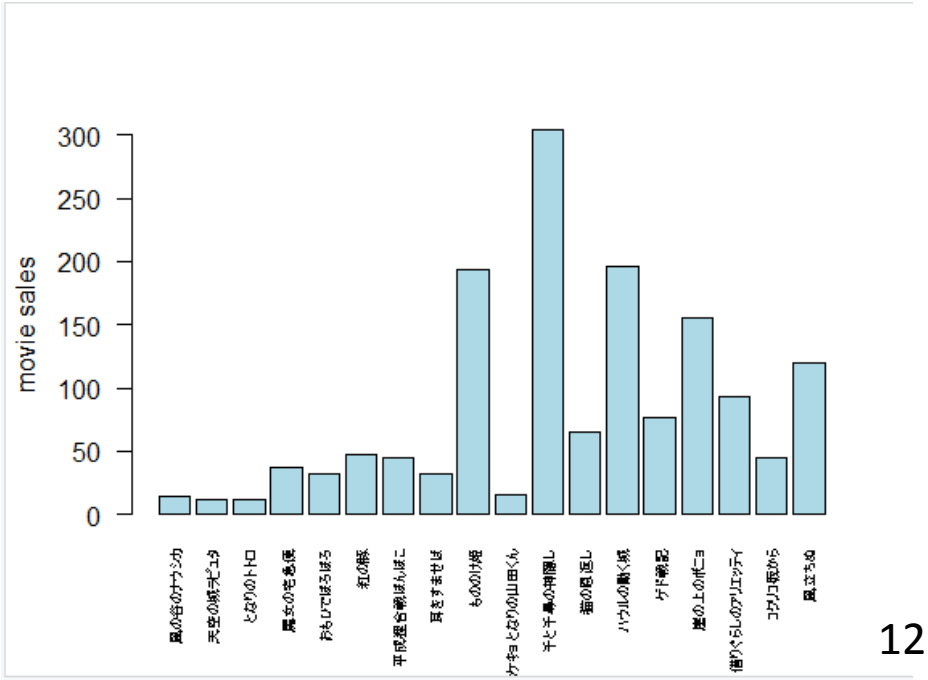
Bar graph

```
#棒グラフ/Bar graph
barplot((data$sales),
        ylab = "movie sales",          #y軸の名前
        col="lightblue",                #色も指定できる
        names.arg = (data$title),       #x軸の名前を指定
        cex.names = 0.5,                #x軸のラベルの文字の大きさ
        las=2                           #x軸のラベルを縦書きに
)
```

barplot(data)

	title	year	sales	audience
2	風の谷のナウシカ	1984	14.8	91
3	天空の城ラピュタ	1986	11.6	77
4	となりのトトロ	1988	11.7	80
5	魔女の宅急便	1989	36.5	264
6	おもひでぽろぽろ	1991	31.8	216
7	紅の豚	1992	47.6	304
8	平成狸合戦ぽんぽこ	1994	44.7	325
9	耳をすませば	1995	31.5	208
10	もののけ姫	1997	193	1420
11	ホーホケキョ となりの山田くん	1999	15.6	115
12	千と千尋の神隠し	2001	304	2350
13	猫の恩返し	2002	64.6	550
14	ハウルの動く城	2004	196	1500
15	ゲド戦記	2006	76.5	588
16	崖の上のポニョ	2008	155	1200
17	借りぐらしのアリエッティ	2010	92.5	750
18	コクリコ坂から	2011	44.6	355
19	風立ちぬ	2013	120	1000

Y axis

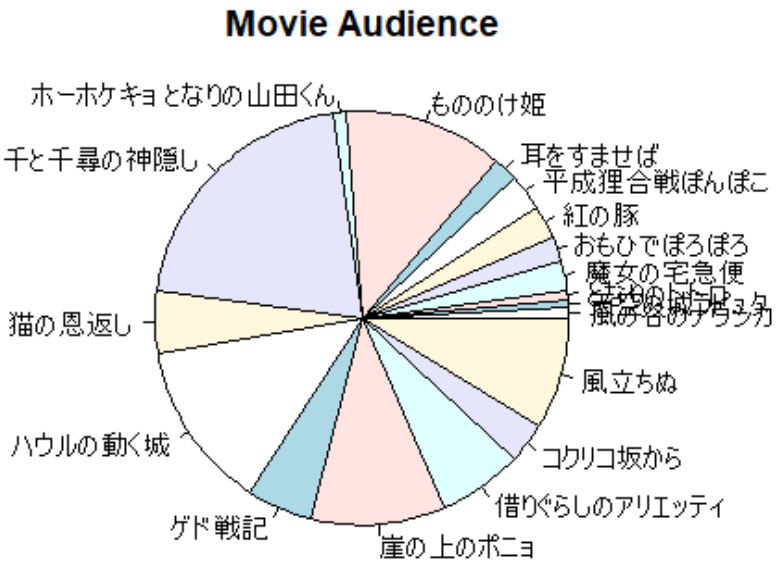


Pie chart

```
77
78  #円グラフ/Pie chart
79  pie(data$audience,
80      main="Movie Audience",          #タイトル
81      labels = (data$title),          #ラベルを入れる
82      radius=1)                      #円の大きさ
83
84
```

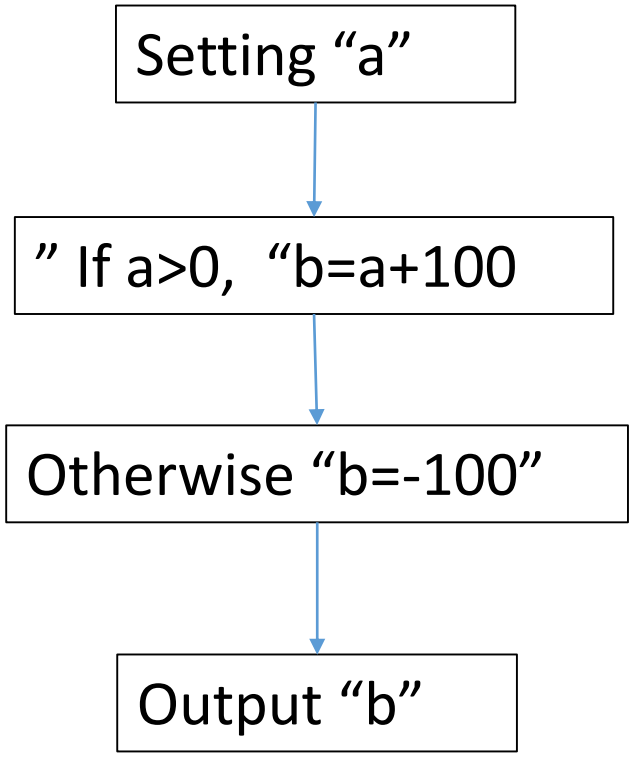
pie(data)

1	title	year	sales	audience
2	風の谷のナウシカ	1984	14.8	91
3	天空の城ラピュタ	1986	11.6	77
4	となりのトトロ	1988	11.7	80
5	魔女の宅急便	1989	36.6	264
6	おもひでぽろぽろ	1991	31.8	216
7	紅の豚	1992	47.6	304
8	平成狸合戦ぽんぽこ	1994	44.7	325
9	耳をすませば	1995	31.6	208
10	もののけ姫	1997	193	1420
11	ホーホケキョ となりの山田くん	1999	15.6	115
12	千と千尋の神隠し	2001	304	2350
13	猫の恩返し	2002	64.6	550
14	ハウルの動く城	2004	196	1500
15	ゲド戦記	2006	76.5	588
16	崖の上のポニョ	2008	155	1200
17	借りぐらしのアリエッティ	2010	92.5	750
18	コクリコ坂から	2011	44.6	355
19	風立ちぬ	2013	120	1000



If文

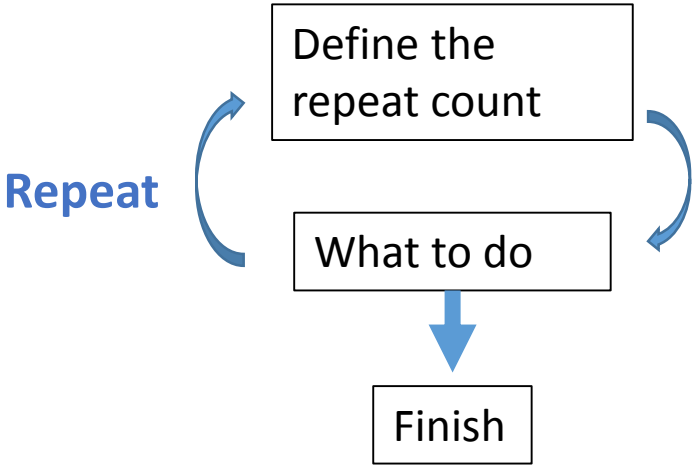
```
89
90  #if文/
91  a=2                  #aを指定
92  if(a>0){             #条件
93    b<-a+100           #当てはまった時
94  }else{               #当てはまらなかったとき
95    b<-(-100)
96  }
97  b                    #bを出力
98
99
```



if (条件式 : conditional expression){
　　実行すること: what to do
} else if (条件式2: conditional expression 2){
　　実行すること
　　.
　　.
}else {
　　実行すること
}

for文

```
 99
100  #for文
101  x<-matrix(1:100,20,5)     #行列を作る
102  y<-matrix(0,20,1)         #
103   for(i in 1:20){          #繰り返し回数
104     y[i]<-max(x[i,])       #各行の最大値をyに代入
105   }
106  y
107
108
```




```
for (i in 1:100){
```

Repeat count

```
y[i]  <-  max(x[I, ]  )
```

What to do

```
}
```

max

1	21	41	61	81
2	22	42	62	82
3	23	43	63	83
:	:	:	:	
18	38	58	78	98
19	39	59	79	99
20	40	60	80	100

Application

Evaluate movies

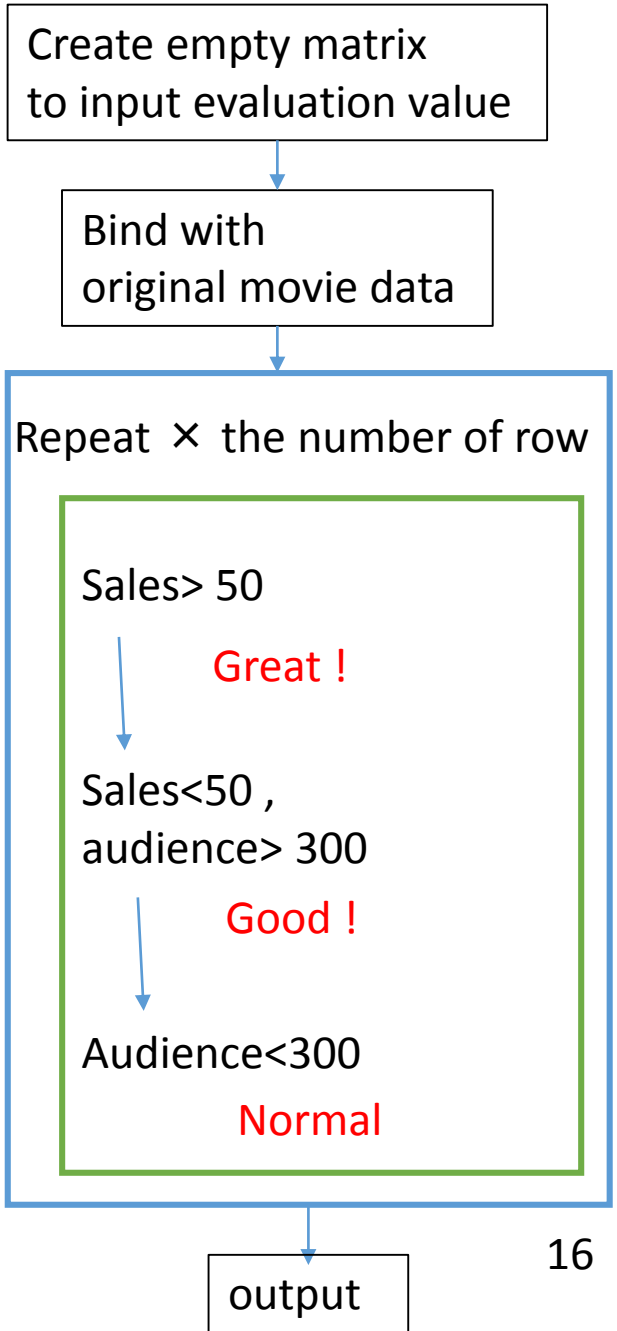
Create matrix

Read csv file and bind

for 文

If 文

```
113
114
115 #空データの作成/Creating an empty data
116 evaluation<-matrix(0:0,nrow=18,ncol=1)
117 #ジブリデータとの結合/ Binding of the movie data
118 new_data<-cbind(data,evaluation)
119
120
121 for(i in 1:nrow(data)){     #dataの行数だけ繰り返す
122   if(data[i,3]>50){         #売り上げが50より大きかった時
123     new_data[i,5]<-as.vector("great")     #great!!
124   }else if(data[i,4]>300){  #売り上げはだめでも，人はたくさん来てるとき
125     new_data[i,5]<-as.vector("good")      #good!
126   }else{                     #それでも
127     new_data[i,5]<-as.vector("normal")  #ジブリはいい映画だ．
128   }
129 }
130
131
132 #csvファイルの出力/Write the csv file
133 write.csv(new_data,"movie_evaluation.csv")
134
```



Application

Evaluation value was added

	title	year	sales	audience	evaluation
1	風の谷のナウシカ	1984	14.8	91	normal
2	天空の城ラピュタ	1986	11.6	77	normal
3	となりのトトロ	1988	11.7	80	normal
4	魔女の宅急便	1989	36.5	264	normal
5	おもひでぽろぽろ	1991	31.8	216	normal
6	紅の豚	1992	47.6	304	good
7	平成狸合戦ぽんぽこ	1994	44.7	325	good
8	耳をすませば	1995	31.5	208	normal
9	もののけ姫	1997	193.0	1420	great
10	ホーホケキョとなりの山田くん	1999	15.6	115	normal
11	千と千尋の神隠し	2001	304.0	2350	great
12	猫の恩返し	2002	64.6	550	great
13	ハウルの動く城	2004	196.0	1500	great
14	ゲド戦記	2006	76.5	588	great
15	崖の上のポニョ	2008	155.0	1200	great
16	借りぐらしのアリエッティ	2010	92.5	750	great

Write csv file

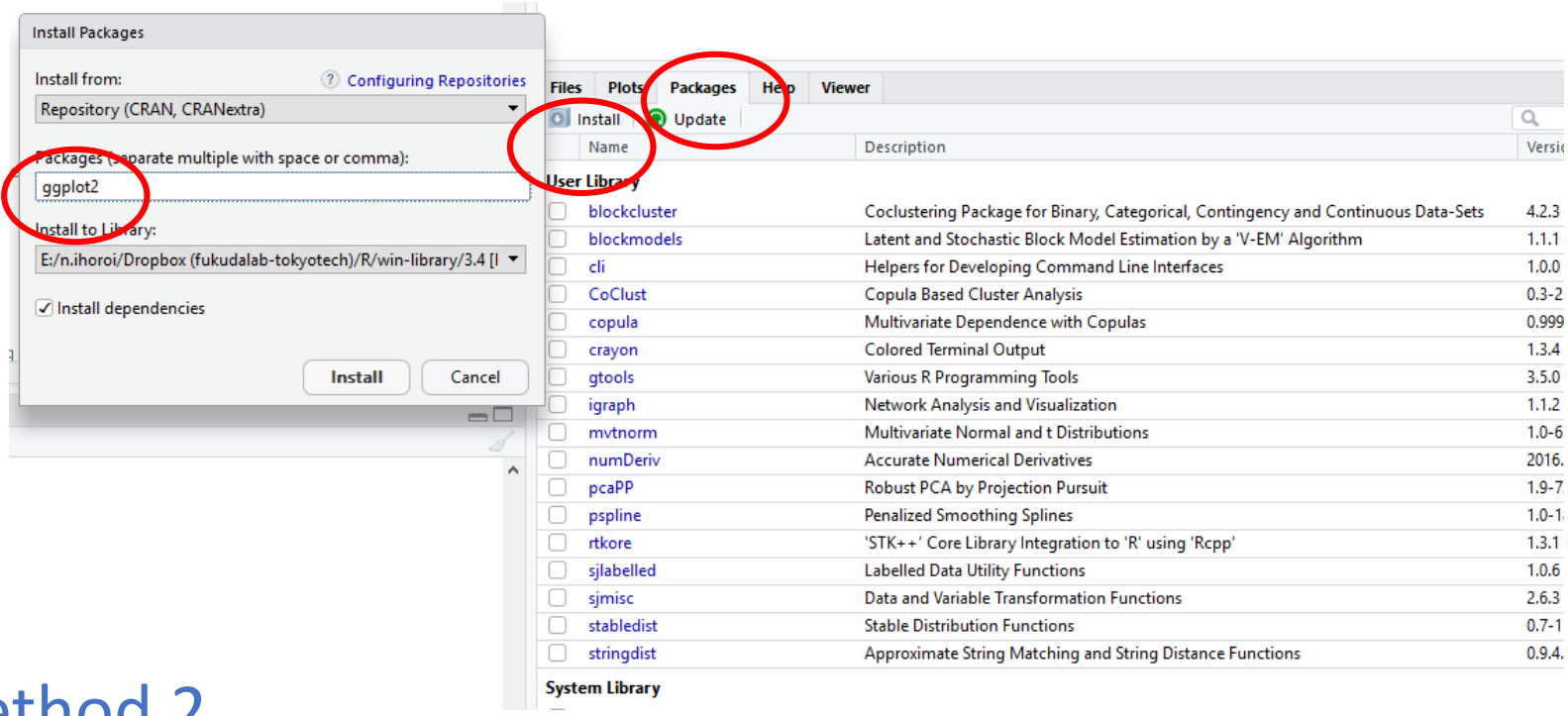
```
131
132  #csvファイルの出力/Write the csv file
133  write.csv(new_data,"movie_evaluation.csv")
134
```

write.csv(data, “file name”)

How to install package

Method 1

1. Click “Package” in the bottom right
2. Click “install” and type “packages name”



Method 2

“install.packages(“package name”)

```
2 install.packages("ggplot2")
```

How to use package

```
141  #dplyr/ use of "dplyr" package
142  library(ggplot2)
143  library(dplyr)
144
```

library (“package name”)

ggplot2

A graphic drawing package
in which beautiful graphs can be drawn

dplyr

Package specialized for data frame
It can process at high spped because written in C++

dplyr

Okinawa.csv

	Observed_Date	macid	site
2	2017/8/6	9179	37
3	2017/8/6	33228	36
4	2017/8/6	52656	35
5	2017/8/6	53506	37
6	2017/8/6	93146	35
7	2017/8/6	108201	35
8	2017/8/6	108201	37
9	2017/8/6	109102	37
10	2017/8/6	123844	35
11	2017/8/6	135300	35
12	2017/8/6	144454	37
13	2017/8/6	173205	37
14	2017/8/6	185798	37

Siteinfo.csv

site	sitename
35	首里城
36	美ら海水族館
37	国際通り

```
146 #csvの読み込み/Read the csv file
147 data2<-read.csv("okinawa.csv")
148 siteinfo<-read.csv("siteinfo.csv")
149
150 #観光地を3か所に限定/Filtering of tourism spot
151 data2%>%
152   filter((site==35)|(site==36)|(site==37))->data3
153
```

filter

Narrow doan the rows that matched the conditional expression

Filtering 3 sites “Churaumi”, ”Syuri-castle and “Kokusai-dori”

dplyr

```
153
154
155 #日付別の各観光地の観光人数/The number of tourist per day and per site
156 data3%>%
157   arrange(Observed_Date,site)%>%     #データの並び替え
158   group_by(Observed_Date,site)%>%    #データのグループ化
159   summarise(np=n())->test2           #集計
160
161 #観光地名をつける/Add a tourism sitename
162 left_join(test2,siteinfo,by="site")->data3  #同じsite番号のところにsitenameを補う
163
```

arrange

Sort function

	Observed_Date	macid	site
1	2017/8/6	9	37
2	2017/8/6	33	36
3	2017/8/6	5	35
4	2017/8/6	5	37
5	2017/8/6	93	35
6	2017/8/6	10	35
7	2017/8/6	108	37
8	2017/8/6		37
9	2017/8/6	123	35
10	2017/8/6	135300	35

Order by date

group_by

Grouping function

	Observed_Date	macid	site
1	2017/8/6	9179	37
2	201		36
3	201	2017/8/6	35
4	201		37
5	2017/8/6	93146	35
6	2017/8/6	108201	35
7	201		37
8	201	2017/8/7	37
9	2017/8/6	123844	35
10	2017/8/6	135300	35

summrise

Aggregation function

	Observed_Date	site	np
1	2017/8/10	35	2
2	2017/8/10	36	2
3	2017/8/10	37	5
4	2017/8/6	35	72
5	2017/8/6	36	29
6	2017/8/6	37	108
7	2017/8/7	35	6
8	2017/8/7	36	33
9	2017/8/7	37	12

left_join

	Observed_Date	site	np	sitename
1	2017/8/10	35	2	首里城
2	2017/8/10	36	2	美ら海水族館
3	2017/8/10	37	5	国際通り
4	2017/8/6	35	72	首里城
5	2017/8/6	36	29	美ら海水族館
6	2017/8/6	37	108	国際通り
7	2017/8/7	35	6	首里城
8	2017/8/7	36	33	美ら海水族館
9	2017/8/7	37	12	国際通り

ggplot

If you change here, you can draw other graphs

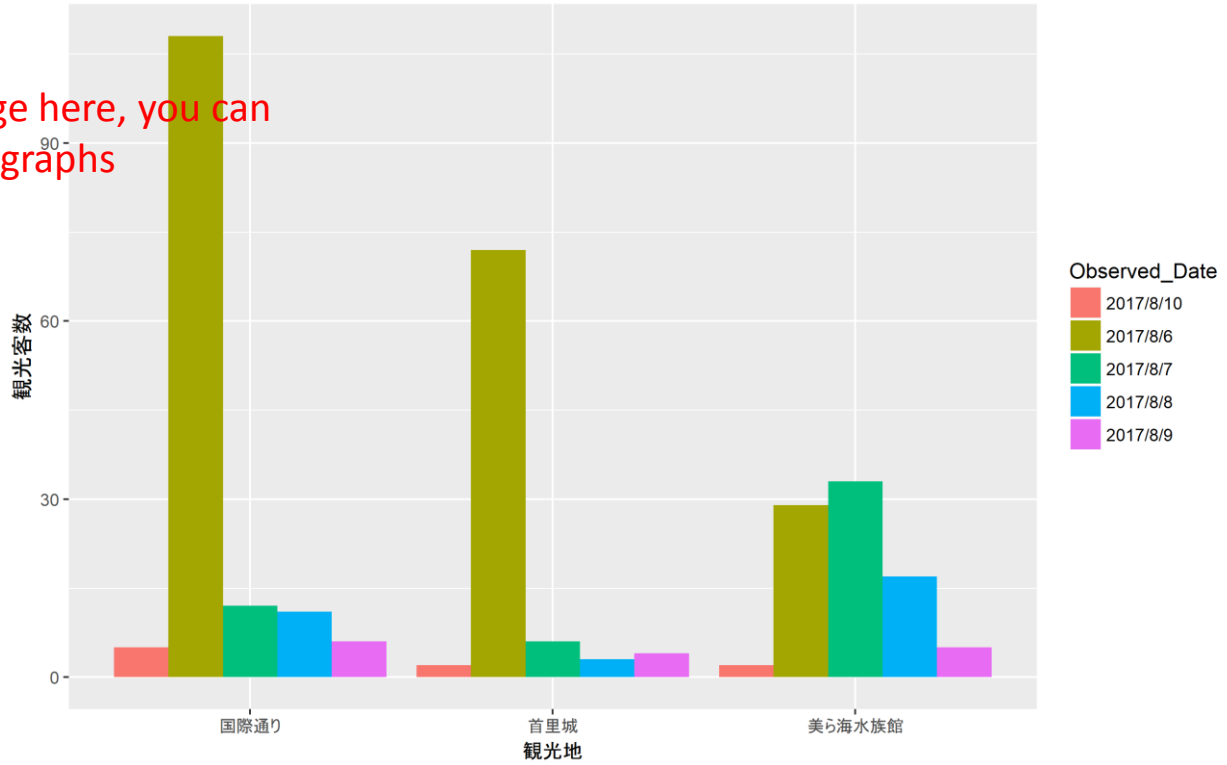
g<-ggplot()+ geom_bar

visualize

Plot(g)

save

ggsave(file="file name", plot=p)




```
165
166  #ggplot /Use of "ggplot" package
167  library(ggplot2)
168
169  g<-ggplot(data3,aes(x=sitename,y=np))+          #データの指定
170    xlab("観光地")+ylab("観光客数")+
171    geom_bar(stat="identity",aes(fill=Observed_Date),position = "dodge")  #日付ごとに塗り分
172
173  plot(g)  #描いてみる
174
175  ggsave(file="施設ごとの観光客数.png",plot = g)
176
177
```