2022年9月22日@京都大学

KUDH Basics
統計ソフトウェア「R」ワークショップ

2日目

（前半）tidyverseでデータフレームを操作しよう

講師：山崎 大暉

（立命館大学・日本学術振興会）

今日の流れと到達目標

1. tidyverseパッケージでデータフレームを操作しよう

- データフレームから行や列を抽出する
- データを変数でグループ化し，要約して新しいデータフレームを作る

2. 図を描画して保存しよう

- ggplot()関数を使ってデータフレームからグラフを作成する
  - 棒グラフ，折れ線グラフ，箱ひげ図，エラーバー付き
- 図の体裁，軸や目盛りを自由に調整する
- 複数の図をまとめる
- 作った図を書き出す

Rのパッケージを使って
データフレームを操作しよう

データフレームの型：ワイド型とロング型

ワイド型データフレーム

地点	6時	12時	18時
札幌			
東京			
福岡			

ロング型データフレーム

地点	時刻	天気
札幌	6時	
札幌	12時	
札幌	18時	
東京	6時	
東京	12時	
東京	18時	
福岡	6時	
福岡	12時	
福岡	18時	

データフレームの型：ワイド型とロング型

ワイド型データフレーム

地点	6時	12時	18時
札幌			
東京			
福岡			

1つの観測

ロング型データフレーム

地点	時刻	天気
札幌	6時	
札幌	12時	
札幌	18時	
東京	6時	
東京	12時	
東京	18時	
福岡	6時	
福岡	12時	
福岡	18時	

1つの観測

Rではロング型が扱いやすい

- ggplot2やdplyrの関数はロング型に対応
  - 第１引数にデータフレームを入れ，列名で変数を指定

    > ggplot(**iris**, aes(x = Sepal.Length, y = Sepal.Width)) +…

  - 列名を指定してデータを抽出する

    > select(iris, **Sepal.Length**, **Species**)

    > filter(iris, **Sepal.Length > 7**)

Rにデフォルトで用意されている
データフレーム「iris」

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

データフレーム iris

- **3種のアヤメのがく片・花びらの長さ・幅を計測したデータ**
  - setosa・versicolor・virginicaが各50個体ずつ（計150個体）
  - Rの練習や解説でよく使われるデータフレーム

- データフレームの頭6行を表示するhead()関数で見てみる
  ※単にirisとだけ打つと150行全部が表示される

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

データフレーム iris

```
> head(iris)
```

行番号（各観測）	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

がく片の長さ　がく片の幅　花びらの長さ　花びらの幅　種

tidyverseパッケージ

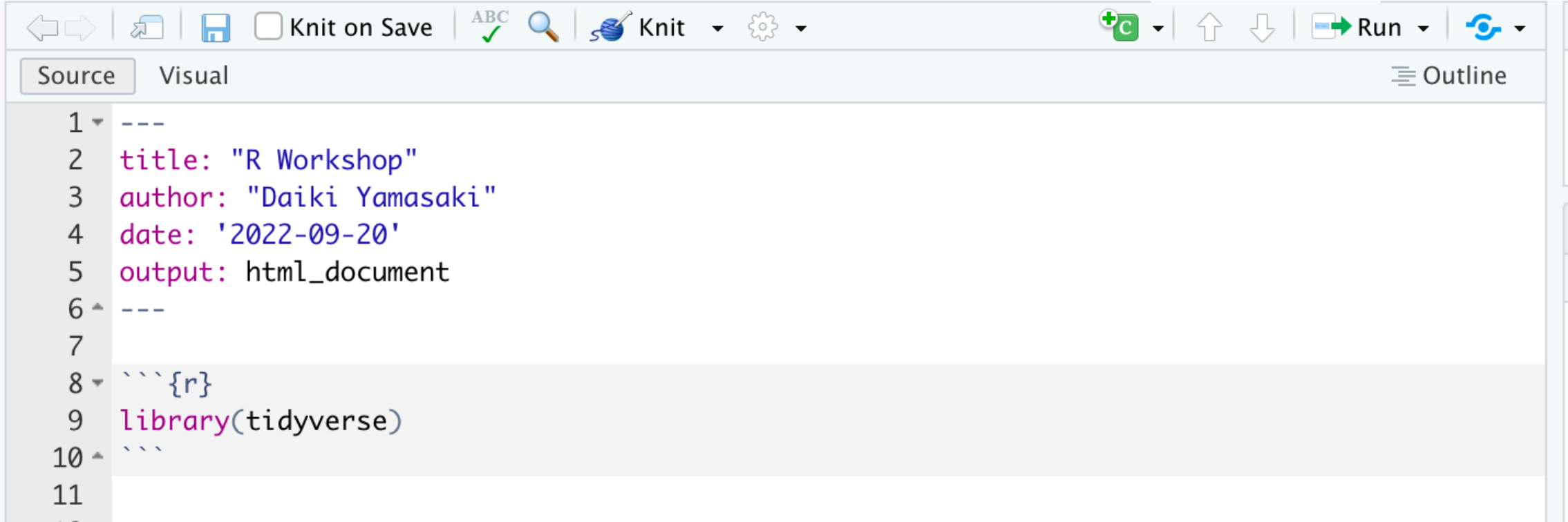
- dplyr：データ操作
- ggplot2：グラフ描画
- magrittr：パイプ演算子%>%
- tidyr：整然データ作成
- readr：データ読み込み
- stringr：文字列操作
- purrr：繰り返し計算
- などなど


```
> install.packages("tidyverse")
```

```
```{r}
library(tidyverse)
```
```

tidyverseパッケージ

```
> install.packages("tidyverse")
```



dplyr

データフレームを操作・加工する関数が詰まったパッケージ

①行の一部を取り出す	filter()関数
②列の一部を取り出す	select()関数
③新しい列を追加する	mutate()関数
④データをグループ化する	group_by()関数
⑤データを要約する	summarise()関数

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

dplyrの関数を使って
データフレームirisを
加工してみよう

①行の一部を抽出する： filter()関数

- filter()関数
  - 第１引数：データフレームの名前
  - 第２引数以降：抽出したい行を表す条件式

データフレームirisからsetosaのアヤメのデータだけ抽出しよう

```
```{r}
filter(iris, Species == "setosa")
```
```

※文字列の指定には ”” ダブルクォーテーションが必要

Rの比較演算子

- 条件式を書く際に使用する記号

記号	意味
==	等しい　※「=」は代入
!=	等しくない
>=	以上
>	より大きい
<=	以下
<	より小さい
%in%	グループメンバシップ

記号	意味
is.na	NA(欠損値)である
!is.na	NA(欠損値)でない
&	かつ
|	または

①行の一部を抽出する： filter()関数

irisからSepal.Lengthが7より大きいデータだけ抽出しよう

```
filter(iris, Sepal.Length > 7) # Sepal.Lengthが7より大きい行を抽出
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1            7.1         3.0          5.9         2.1 virginica
## 2            7.6         3.0          6.6         2.1 virginica
## 3            7.3         2.9          6.3         1.8 virginica
## 4            7.2         3.6          6.1         2.5 virginica
## 5            7.7         3.8          6.7         2.2 virginica
## 6            7.7         2.6          6.9         2.3 virginica
## 7            7.7         2.8          6.7         2.0 virginica
```

①行の一部を抽出する： filter()関数

filter()関数で複数の条件を指定しよう
「Sepal.Lengthが6.5以上 かつ Speciesがversicolor」の行を抽出

※条件式を ＆ または コンマ で区切る　⇒「かつ」を表す

```
 # Sepal.Lengthが6.5以上かつSpeciesが"versicolor"である行を抽出
filter(iris, Sepal.Length >= 6.5, Species == "versicolor")
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1          7.0         3.2          4.7         1.4 versicolor
## 2          6.9         3.1          4.9         1.5 versicolor
## 3          6.5         2.8          4.6         1.5 versicolor
## 4          6.6         2.9          4.6         1.3 versicolor
## 5          6.7         3.1          4.4         1.4 versicolor
```

①行の一部を抽出する： filter()関数

filter()関数で複数の条件を指定しよう
「Sepal.Lengthが7以上 または Sepal.Lengthが4以上 」の行を抽出

※条件式を｜（縦線）で区切る　⇒「または」を表す

```
 # Sepal.Lengthが7以上 またはSepal.Lengthが4以上である行を抽出
filter(iris, Sepal.Length >= 7| Sepal.Width >= 4)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1           5.8         4.0          1.2         0.2     setosa
## 2           5.7         4.4          1.5         0.4     setosa
## 3           5.2         4.1          1.5         0.1     setosa
## 4           5.5         4.2          1.4         0.2     setosa
## 5           7.0         3.2          4.7         1.4 versicolor
```

②列の一部を抽出する： select()関数

- select()関数
  - 第１引数：データフレームの名前
  - 第２引数以降：抽出したい列名

irisからSpeciesという名前の列だけ抽出しよう

```
select(iris, Species)
```

irisからSpecies, Sepal.Length, Sepal.Widthの列を抽出しよう

```
select(iris, Species, Sepal.Length, Sepal.Width)
```

②列の一部を抽出する： select()関数

指定した列を除外しよう

```
select(iris, -Species)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4

Sで始まる列名だけ抽出しよう

```
select(iris, starts_with("S"))
```

	Sepal.Length	Sepal.Width	Species
1	5.1	3.5	setosa
2	4.9	3.0	setosa
3	4.7	3.2	setosa
4	4.6	3.1	setosa
5	5.0	3.6	setosa
6	5.4	3.9	setosa

③新しい列を追加する：mutate()関数

- mutate()関数
  - ・第１引数：データフレーム
    第２引数以降：新しい列の名前と追加する内容を指定

各アヤメのがく・花びらの面積を計算し，新規な列に追加しよう

```
mutate(iris,
       Area_Sepal = Sepal.Length * Sepal.Width,
       Area_Petal = Petal.Length * Petal.Width)
```

新規な列名を指定

それぞれの列に，各行の計算結果を代入

Sepal.Length		Sepal.Width	Petal.Length		Petal.Width	Species	Area_Sepal	Area_Petal
5.1	×	3.5	1.4	×	0.2	setosa	17.85	0.28
4.9		3.0	1.4		0.2	setosa	14.70	0.28
4.7		3.2	1.3		0.2	setosa	15.04	0.26
4.6		3.1	1.5		0.2	setosa	14.26	0.30
5.0		3.6	1.4		0.2	setosa	18.00	0.28
5.4		3.9	1.7		0.4	setosa	21.06	0.68

③条件ごとに異なる値を追加する

- mutate()関数 + if_else()関数
  - if_else(条件式, TRUEのとき, FALSEのとき)

Speciesがvirginicaならば”virginica”，それ以外は”NOT virginica”を入れる列Species2を追加しよう

```
mutate(iris,
       Species2 = if_else(Species == "virginica", "virginica", "NOT virginica"))
```

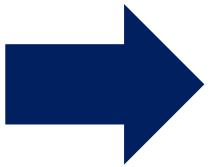
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species         Species2
## 1            5.1         3.5          1.4         0.2     setosa    NOT virginica
## 2            4.9         3.0          1.4         0.2     setosa    NOT virginica
## 3            4.7         3.2          1.3         0.2     setosa    NOT virginica
## 4            4.6         3.1          1.5         0.2     setosa    NOT virginica
## 5            5.0         3.6          1.4         0.2     setosa    NOT virginica
## 6            5.4         3.9          1.7         0.4     setosa    NOT virginica
## 7            4.6         3.4          1.4         0.3     setosa    NOT virginica
```

③変数の水準のラベルを変更する

- mutate()関数 + fct_recode()関数
  - mutate(新しい変数名 = fct_recode(元の変数名,
    新しいラベル1 = “元のラベル1”,
    新しいラベル2 = “元のラベル2”,
    …)

irisのSpeciesの3種類を新しい名前にしよう

```
mutate(iris,
       Ayame = fct_recode(Species,
                          セトサ = "setosa",
                          バージニカ = "virginica",
                          バージカラー = "versicolor")
       )
```



```
tal.Width Species  Ayame
      0.2  setosa セトサ
      0.2  setosa セトサ
      0.2  setosa セトサ
      0.2  setosa セトサ
      0.2  setosa セトサ
      0.4  setosa セトサ
```

Ayameという列を作り，元のラベルを新しいラベルに変更した

④データを一行に要約： summarise()関数

- summarise()関数
  - 第１引数：データフレーム
    第２引数以降：要約して出力するデータフレームの列の情報

**irisのSepal.Lengthの平均と不偏標準偏差を計算し，
その結果を新たなデータフレームに要約しよう**

```
summarise(iris,
          Mean = mean(Sepal.Length),
          SD = sd(Sepal.Length))
```

```
##       Mean        SD
## 1 5.843333 0.8280661
```

新規な列名 = 関数名(計算したい元の列名)

⑤データをグループ化する：group_by()関数

- group_by()関数
  - 第１引数：データフレーム
    第２引数以降：グループ化する変数名

irisのデータをSpeciesごとにグループ化しよう

```
group_by(iris, Species)
```

- Speciesでグループ化したirisを，**iris_g**というオブジェクトに代入
- iris_gの頭6行だけを表示

```
iris_g <- group_by(iris, Species)
head(iris_g)
```

変数Speciesでグループ化されていることを表す

```
## # A tibble: 6 × 5
## # Groups:   Species [1]
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##          <dbl>       <dbl>        <dbl>       <dbl> <fct>
## 1          5.1         3.5          1.4         0.2 setosa
## 2          4.9         3            1.4         0.2 setosa
## 3          4.7         3.2          1.3         0.2 setosa
## 4          4.6         3.1          1.5         0.2 setosa
## 5          5           3.6          1.4         0.2 setosa
## 6          5.4         3.9          1.7         0.4 setosa
```

group_by()関数+summarise()関数

グループごとにデータを要約しよう

- group_by()関数でグループ化したデータフレームiris_gを，summarise()関数の第１引数に渡す

```
iris_g <- group_by(iris, Species)

summarise(iris_g,
          Mean = mean(Sepal.Length),
          SD = sd(Sepal.Length))
```

SpeciesごとにMeanとSDが計算されている

```
# A tibble: 3 × 3
  Species      Mean    SD
  <fct>       <dbl> <dbl>
1 setosa       5.01 0.352
2 versicolor   5.94 0.516
3 virginica    6.59 0.636
```

データフレームToothGrowth

モルモットにビタミンCかジュースを投与し，歯の長さ(len)を記録

- 投与物(supp)：ビタミンC（VC） / オレンジジュース（OJ）の2水準
- 投与量(dose)：0.5 / 1.0 / 2.0 mgの3水準

```
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

ToothGrowthをdoseとsuppの2変数でグループ化しよう

- group_by()関数
  - 第１引数：データフレーム
  - 第２引数以降：グループ化する変数名

```
TG_g <- group_by(ToothGrowth, dose, supp)
head(TG_g)
```

一旦別のデータフレームに格納

頭6行だけを表示

```
# A tibble: 6 × 3
# Groups:   dose, supp [1]
    len supp   dose
  <dbl> <fct> <dbl>
1   4.2 VC      0.5
2  11.5 VC      0.5
3   7.3 VC      0.5
4   5.8 VC      0.5
5   6.4 VC      0.5
6  10   VC      0.5
```

変数doseと変数suppの2変数でグループ化された

group_by()関数+summarise()関数

- 2変数でグループ化したデータフレームTG_gをsummarise()関数に渡し，条件ごとに要約統計量を計算する

```
TG_g <- group_by(ToothGrowth, dose, supp)

summarise(TG_g,
          Mean = mean(len),
          Median = median(len),
          Variance = var(len),
          SD = sd(len))
```

```
# A tibble: 6 × 6
# Groups:   dose [3]
   dose supp   Mean Median Variance    SD
  <dbl> <fct> <dbl>  <dbl>    <dbl> <dbl>
1   0.5 OJ     13.2   12.2     19.9  4.46
2   0.5 VC      7.98   7.15     7.54 2.75
3   1   OJ     22.7   23.5     15.3  3.91
4   1   VC     16.8   16.5      6.33 2.52
5   2   OJ     26.1   26.0      7.05 2.66
6   2   VC     26.1   26.0     23.0  4.80
```

dose（3水準）×supp（2水準）ごとに
平均値，中央値，不偏分散，不偏標準偏差
が計算されている

group_by()関数+mutate()関数

- グループごとに新しい列の値を追加する
  - group_by()関数で出力したデータフレームをmutate()関数の第１引数へ

```
iris_g <- group_by(iris, Species)
mutate(iris_g,
       Mean = mean(Sepal.Length)) # 平均値
```

```
## # A tibble: 150 x 6
## # Groups:   Species [3]
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species  Mean
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>   <dbl>
##  1          5.10        3.50         1.40       0.200 setosa   5.01
##  2          4.90        3.00         1.40       0.200 setosa   5.01
##  3          4.70        3.20         1.30       0.200 setosa   5.01
##  4          4.60        3.10         1.50       0.200 setosa   5.01
##  5          5.00        3.60         1.40       0.200 setosa   5.01
##  6          5.40        3.90         1.70       0.400 setosa   5.01
```

ungroup()関数：グループ化を解除

- ungroup()関数
  - 第１引数：グループ化されたデータフレーム

```
iris_g <- group_by(iris, Species)
ungroup(iris_g)
```

グループ化されていない

```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.10        3.50         1.40       0.200 setosa
##  2          4.90        3.00         1.40       0.200 setosa
##  3          4.70        3.20         1.30       0.200 setosa
##  4          4.60        3.10         1.50       0.200 setosa
##  5          5.00        3.60         1.40       0.200 setosa
##  6          5.40        3.90         1.70       0.400 setosa
```

パイプ演算子を使ったパイプ処理

パイプ演算子（Win: Ctrl+Shift+M，Mac: ⌘+Shift+M）

- tidyverse内のmagrittrパッケージで扱える
- 前の値を，後ろの関数の第一引数として渡す

```
%>%
```

- 上下のコマンドはそれぞれ同じ意味になる

```
head(iris)

iris %>% head()
```

```
filter(iris, Species == "setosa")

iris %>% filter(Species == "setosa")
```

パイプ演算子を使ったパイプ処理

パイプ演算子

- tidyverseの関数は第1引数がデータフレーム
- 関数間のデータフレームの受け渡しがスムーズ
- いちいちデータフレームを作る必要がない
- 複数の処理をパイプライン化してひと繋がりにできる

以下のコマンドをパイプ演算子を使って書いてみよう

① irisのうちSepal.Lengthが5より大きい行を抽出する
② Speciesごとにグループ化する
③ グループごとにSepal.LengthのMeanとSDを計算する

```
iris_2 <- filter(iris, Sepal.Length > 5)

iris_g <- group_by(iris_2, Species)

summarise(iris_g,
          Mean = mean(Sepal.Length),
          SD = sd(Sepal.Length))
```

```
iris %>%
  filter(Sepal.Length > 5) %>%
  group_by(Species) %>%
  summarise(Mean = mean(Sepal.Length),
            SD = sd(Sepal.Length))
```

dplyrでデータフレームを操作

紹介したdplyrの関数

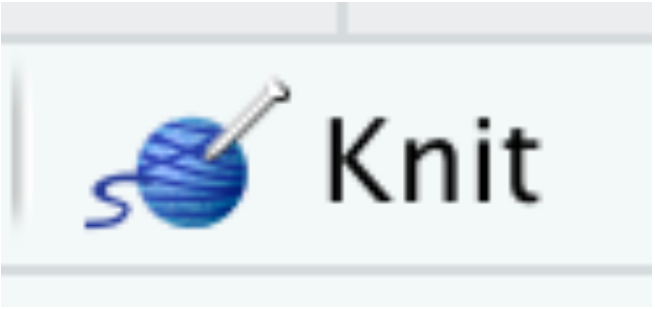
- 行の一部を取り出す　filter()関数
- 列の一部を取り出す　select()関数
- 新しい列を追加する　mutate()関数
- データをグループ化　group_by()関数
- データを要約する　summarise()関数

関数の合わせ技

- group_by() + summarise()
- group_by() + mutate()
- mutate() + if_else()

その他Tips

- 行を並び替える　arrange()関数
- 列名を変える　rename()関数
- 欠損値を除外　filter(df, !is.na(列名))
- パイプ演算子　%>%



R Markdownをknitして
htmlを出力しましょう