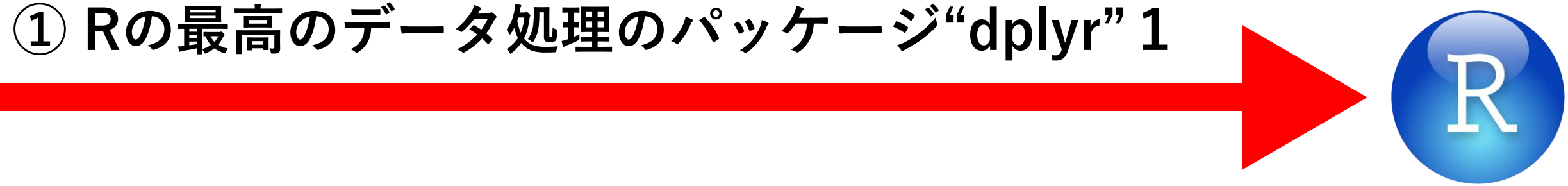
# 有用なR分析技術

① Rの最高のデータ処理のパッケージ“dplyr” 1



# dplyrを使う理由は？

**①強力な全処理機能！**
→整列、フィルタリング、カラムの選択など簡単にできる

**②一般的なR関数より直観的で使いやすい！**
→select 、arrange 、filterなど関数名でどのような機能を持っているのか簡単に把握できる

**③コードの整理が簡単！**
→パイプ記号（%>%）で簡単に整理することもできる（Ctrl＋Shift＋M）

**④他のtidyverseパッケージとの連動も簡単！**
→ggplot2などの強力な機能を持つパッケージと同じtidyverseパッケージのため連動が簡単

# 基礎となるdplyrの関数

**①filter()**
→提示した条件と一致している資料を表示する

**②arrange()**
→行（Row）を整列する

**③select()**
→特定な列（Column）を選択する

**④mutate()**
→新しい変数を計算・生成する

**⑤summarize()**
→データの統計量を計算し、簡略に表記する

# 練習に使うデータの紹介

## 以下はとある大学の授業の成績表である

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Student Number | Name | Major | Midterm | Final |
| 2 | 1301 | Choi | Literature | 50 | 45 |
| 3 | 1302 | Park | Engineering | 76 | 42 |
| 4 | 1303 | Han | Literature | 100 | 98 |
| 5 | 1304 | Jin | Engineering | 85 | 92 |
| 6 | 1305 | Liu | Medicine | 100 | 100 |
| 7 | 1306 | Li | Engineering | 86 | 100 |
| 8 | 1307 | Yamaoka | Social Science | 75 | 90 |
| 9 | 1308 | Sirasaki | Natural Science | 88 | 90 |
| 10 | 1309 | Honda | Natural Science | 92 | 88 |
| 11 | 1310 | Yamada | Social studies | 45 | 0 |
| 12 | 1311 | Nguyen | Literature | 100 | 92 |
| 13 | 1312 | Lee | Medicine | 85 | 77 |
| 14 | 1313 | Hong | Engineering | 82 | 90 |
| 15 | 1314 | Hua | Natural Science | 56 | 42 |
| 16 | 1315 | Ma | Natural Science | 87 | 78 |
| 17 | 1316 | Okusora | Literature | 75 | 62 |
| 18 | 1317 | Tu | Engineering | 83 | 55 |
| 19 | 1318 | Satou | Social studies | 90 | 82 |

| | Student Number | Name | Major | Midterm | Final |
|---|---|---|---|---|---|
| 1 | 1301 | Choi | Literature | 50 | 45 |
| 2 | 1302 | Park | Engineering | 76 | 42 |
| 3 | 1303 | Han | Literature | 100 | 98 |
| 4 | 1304 | Jin | Engineering | 85 | 92 |
| 5 | 1305 | Liu | Medicine | 100 | 100 |
| 6 | 1306 | Li | Engineering | 86 | 100 |
| 7 | 1307 | Yamaoka | Social Science | 75 | 90 |
| 8 | 1308 | Sirasaki | Natural Science | 88 | 90 |
| 9 | 1309 | Honda | Natural Science | 92 | 88 |
| 10 | 1310 | Yamada | Social studies | 45 | 0 |
| 11 | 1311 | Nguyen | Literature | 100 | 92 |
| 12 | 1312 | Lee | Medicine | 85 | 77 |
| 13 | 1313 | Hong | Engineering | 82 | 90 |
| 14 | 1314 | Hua | Natural Science | 56 | 42 |
| 15 | 1315 | Ma | Natural Science | 87 | 78 |
| 16 | 1316 | Okusora | Literature | 75 | 62 |
| 17 | 1317 | Tu | Engineering | 83 | 55 |
| 18 | 1318 | Satou | Social studies | 90 | 82 |

# dplyrの関数①：filter()

①filter()関数で条件を付けるためには演算子が必要

→四則演算子（+, -, x, /）

→論理演算子（And：&, Or：|, Not：!,
Same：==, Not same：!=,
大きさの比較（>, < , >=, <=））

②複数の条件を付けて検索することも可能（パイプ記号（%>%）を使うことをおすすめ）

# dplyrの関数①：filter()

レベル1：MajorがEngineeringの受講生を検索せよ

```
> df %>% dplyr::filter(Major == 'Engineering')
# A tibble: 5 x 5
  `Student Number` Name  Major       Midterm Final
             <dbl> <chr> <chr>         <dbl> <dbl>
1             1302 Park  Engineering      76    42
2             1304 Jin   Engineering      85    92
3             1306 Li    Engineering      86   100
4             1313 Hong  Engineering      82    90
5             1317 Tu    Engineering      83    55
```

★添付されている“Example data (Rの最高のデータ処理のパッケージ　”dplyr”)“を参照

# dplyrの関数①：filter()

レベル2：MajorがEngineering以外の受講生を検索せよ

```
> df %>% dplyr::filter(Major ≠ 'Engineering')
# A tibble: 13 × 5
   `Student Number` Name     Major           Midterm Final
              <dbl> <chr>    <chr>             <dbl> <dbl>
 1             1301 Choi     Literature           50    45
 2             1303 Han      Literature          100    98
 3             1305 Liu      Medicine            100   100
 4             1307 Yamaoka  Social Science       75    90
 5             1308 Sirasaki Natural Science      88    90
 6             1309 Honda    Natural Science      92    88
 7             1310 Yamada   Social studies       45     0
 8             1311 Nguyen   Literature          100    92
 9             1312 Lee      Medicine             85    77
10             1314 Hua      Natural Science      56    42
11             1315 Ma       Natural Science      87    78
12             1316 Okusora  Literature           75    62
13             1318 Satou    Social studies       90    82
```

!=と書くと
自動変換

# dplyrの関数①：filter()

レベル3：MajorがLiterature以外で、Midtermが85点以上、Finalが80点以上の受講生を検索せよ

>=と書くと
自動変換

```
> df %>% dplyr::filter(Major ≠ 'Literature' & Midterm ≧ 85 & Final ≧ 80)
# A tibble: 6 × 5
  `Student Number` Name      Major           Midterm Final
             <dbl> <chr>     <chr>             <dbl> <dbl>
1             1304 Jin       Engineering          85    92
2             1305 Liu       Medicine            100   100
3             1306 Li        Engineering          86   100
4             1308 Sirasaki  Natural Science      88    90
5             1309 Honda     Natural Science      92    88
6             1318 Satou     Social studies       90    82
```

# dplyrの関数②：arrange()

①arrange()でAscending（昇順）もしくはDescending（降順）にデータを整列することができる

②特定なColumnを基準に整列することが可能

③重複することも可能

# dplyrの関数②：arrange()

受講生のName基準に昇順整列をせよ

```
> df %>% dplyr::arrange(Name)
# A tibble: 18 × 5
   `Student Number` Name     Major           Midterm Final
              <dbl> <chr>    <chr>             <dbl> <dbl>
 1             1301 Choi     Literature           50    45
 2             1303 Han      Literature          100    98
 3             1309 Honda    Natural Science      92    88
 4             1313 Hong     Engineering          82    90
 5             1314 Hua      Natural Science      56    42
 6             1304 Jin      Engineering          85    92
 7             1312 Lee      Medicine             85    77
 8             1306 Li       Engineering          86   100
 9             1305 Liu      Medicine            100   100
10             1315 Ma       Natural Science      87    78
11             1311 Nguyen   Literature          100    92
12             1316 Okusora  Literature           75    62
13             1302 Park     Engineering          76    42
14             1318 Satou    Social studies       90    82
15             1308 Sirasaki Natural Science      88    90
16             1317 Tu       Engineering          83    55
17             1310 Yamada   Social studies       45     0
18             1307 Yamaoka  Social Science       75    90
```

# dplyrの関数③：select()

①特定なColumnを選択して表示することができる

②複数のColumnを選択したり、特定なカラムを除去して示すことも可能

# dplyrの関数③：select()

## レベル1：Nameカラムを選択せよ

```
> df %>% dplyr::select(Name)
# A tibble: 18 × 1
   Name
   <chr>
 1 Choi
 2 Park
 3 Han
 4 Jin
 5 Liu
 6 Li
 7 Yamaoka
 8 Sirasaki
 9 Honda
10 Yamada
11 Nguyen
12 Lee
13 Hong
14 Hua
15 Ma
16 Okusora
17 Tu
18 Satou
```

# dplyrの関数③：select()

レベル2：NameとMidterm、Finalカラムを選択せよ

```
> df %>% dplyr::select(Name, Midterm, Final)
# A tibble: 18 × 3
   Name     Midterm Final
   <chr>      <dbl> <dbl>
 1 Choi          50    45
 2 Park          76    42
 3 Han          100    98
 4 Jin           85    92
 5 Liu          100   100
 6 Li            86   100
 7 Yamaoka       75    90
 8 Sirasaki      88    90
 9 Honda         92    88
10 Yamada        45     0
11 Nguyen       100    92
12 Lee           85    77
13 Hong          82    90
14 Hua           56    42
15 Ma            87    78
16 Okusora       75    62
17 Tu            83    55
18 Satou         90    82
```

# dplyrの関数③：select()

レベル3：Student Number以外のすべてのカラムを選択せよ

```
> df %>% dplyr::select(!`Student Number`)
# A tibble: 18 × 4
   Name     Major          Midterm Final
   <chr>    <chr>            <dbl> <dbl>
 1 Choi     Literature          50    45
 2 Park     Engineering         76    42
 3 Han      Literature         100    98
 4 Jin      Engineering         85    92
 5 Liu      Medicine           100   100
 6 Li       Engineering         86   100
 7 Yamaoka  Social Science      75    90
 8 Sirasaki Natural Science     88    90
 9 Honda    Natural Science     92    88
10 Yamada   Social studies      45     0
11 Nguyen   Literature         100    92
12 Lee      Medicine           85    77
13 Hong     Engineering         82    90
14 Hua      Natural Science     56    42
15 Ma       Natural Science     87    78
16 Okusora  Literature          75    62
17 Tu       Engineering         83    55
18 Satou    Social studies      90    82
```

# dplyrの関数④：mutate()

①新しいカラムを追加するとき使う

②すべて同じ内容を含むカラムを入れることも、存在するカラムをベースに計算した結果を追加することもできる

③ifと論理演算子を応用することもできる

④条件が3つ以上の場合はcase_when()関数が便利

# dplyrの関数④：mutate()

レベル2：MidtermとFinalの平均を計算するカラムを作成せよ

```
> df %>% dplyr::mutate(Average = (Midterm+Final)/2)
# A tibble: 18 × 6
   `Student Number` Name     Major           Midterm Final Average
              <dbl> <chr>    <chr>             <dbl> <dbl>   <dbl>
 1             1301 Choi     Literature           50    45    47.5
 2             1302 Park     Engineering          76    42    59
 3             1303 Han      Literature          100    98    99
 4             1304 Jin      Engineering          85    92    88.5
 5             1305 Liu      Medicine            100   100   100
 6             1306 Li       Engineering          86   100    93
 7             1307 Yamaoka  Social Science       75    90    82.5
 8             1308 Sirasaki Natural Science      88    90    89
 9             1309 Honda    Natural Science      92    88    90
10             1310 Yamada   Social studies       45     0    22.5
11             1311 Nguyen   Literature          100    92    96
12             1312 Lee      Medicine             85    77    81
13             1313 Hong     Engineering          82    90    86
14             1314 Hua      Natural Science      56    42    49
15             1315 Ma       Natural Science      87    78    82.5
16             1316 Okusora  Literature           75    62    68.5
17             1317 Tu       Engineering          83    55    69
18             1318 Satou    Social studies       90    82    86
```

# dplyrの関数④：mutate()

レベル4以上！：MidtermとFinalの平均を示す“Average”カラムを作成した後、Averageの数値が60以上は“Pass”、それ以外は“Non-pass”を示す新しいカラムの“Result”を作成せよ

```
> df %>% dplyr::mutate(Average = (Midterm+Final)/2,
+                      Result = ifelse(Average ≥ 60, 'Pass', 'Non-pass'))
# A tibble: 18 × 7
   `Student Number` Name     Major           Midterm Final Average Result
              <dbl> <chr>    <chr>             <dbl> <dbl>   <dbl> <chr>
 1             1301 Choi     Literature           50    45    47.5 Non-pass
 2             1302 Park     Engineering          76    42    59   Non-pass
 3             1303 Han      Literature          100    98    99   Pass
 4             1304 Jin      Engineering          85    92    88.5 Pass
 5             1305 Liu      Medicine            100   100   100   Pass
 6             1306 Li       Engineering          86   100    93   Pass
 7             1307 Yamaoka  Social Science       75    90    82.5 Pass
 8             1308 Sirasaki Natural Science      88    90    89   Pass
 9             1309 Honda    Natural Science      92    88    90   Pass
10             1310 Yamada   Social studies       45     0    22.5 Non-pass
11             1311 Nguyen   Literature          100    92    96   Pass
12             1312 Lee      Medicine             85    77    81   Pass
13             1313 Hong     Engineering          82    90    86   Pass
14             1314 Hua      Natural Science      56    42    49   Non-pass
15             1315 Ma       Natural Science      87    78    82.5 Pass
16             1316 Okusora  Literature           75    62    68.5 Pass
17             1317 Tu       Engineering          83    55    69   Pass
18             1318 Satou    Social studies       90    82    86   Pass
```

# dplyrの関数⑤：summarize()

①数値型データの統計量を計算する

②様々な統計オプションがあるが、以下のオプションが主に使われている
→mean(x, na.rm = TRUE)：平均値の計算（欠損値を含む場合はFALSEに変更）
→n()：データの個数

③group_byを使うことでグループを基準として計算することもできる

④”summarise”と書いてもOK

# dplyrの関数⑤：summarize()

レベル1：Midtermの平均値をsummarize関数を用いて計算せよ

```
> df %>% dplyr::summarize(Average = mean(Midterm))
# A tibble: 1 x 1
  Average
    <dbl>
1    80.8
```

# dplyrの関数⑤：summarize()

レベル3：専攻別MidtermとFinalの平均値を計算せよ

```
> dplyr::summarize(group_by(df, Major),
+                  Midterm_average = mean(Midterm),
+                  Final_average = mean(Final))
# A tibble: 6 × 3
  Major          Midterm_average Final_average
  <chr>                    <dbl>         <dbl>
1 Engineering               82.4          75.8
2 Literature                81.2          74.2
3 Medicine                  92.5          88.5
4 Natural Science           80.8          74.5
5 Social Science            75            90
6 Social studies            67.5          41
```

# 演習問題（挑戦）

以下の条件に合わせてデータを処理しなさい

Medicine専攻以外の理系（Natural ScienceとEngineering）の場合、平均値が85点以上の場合は奨学金の対象になる。 MidtermとFinalの平均値を示すAverageカラムを生成し、該当カラムも示すこと。また、対象学生を示すScholarshipカラムを生成し、対象学生には“S”を、対象外の学生には“ー”を表記すること。最後に、対象学生のStudent number、Name、Major、Averageのみ示すこと。

# 演習問題（挑戦）

```
> df2 ← df %>% dplyr::mutate(Average = (Midterm + Final) / 2) %>%
+   dplyr::mutate(Scholarship =
+                   ifelse(Average ≥ 85 & (Major == 'Engineering'  | Major == 'Natural Science'), "S", "-")) %>%
+   dplyr::filter(Scholarship == 'S') %>%
+   dplyr::select(`Student Number`, Name, Major, Average)
>
> df2
# A tibble: 5 × 4
  `Student Number` Name     Major           Average
             <dbl> <chr>    <chr>             <dbl>
1             1304 Jin      Engineering        88.5
2             1306 Li       Engineering        93
3             1308 Sirasaki Natural Science    89
4             1309 Honda    Natural Science    90
5             1313 Hong     Engineering        86
```