# dplyr包

# 大纲

第一部分 dplyr简介与安装
第二部分 数据预览
第三部分 数据筛选
第四部分 数据重排序
第五部分 数据变形
第六部分 数据汇总
第七部分 数据分组
第八部分 其他技巧

# 第一部分 dplyr简介与安装

- dplyr是plyr包的升级版，可以轻松地对数据进行筛选、变形、汇总、分组、管道等各式各样的数据处理操作，完全可以满足90%以上用户的使用需求。
- 安装：>install.packages(“dplyr”)

# 官方教程讲解

我们所用的数据集是来自nycflights13包里的flights，这也是flight官方例子所用的包。

- 导入包：>library(dplyr)
- 导入数据包：>library(nycflights13)

# 第二部分 数据预览

数据有时候会有很多条，如果一次全部打印，将会花费很多时间，同时也看不到每一行的名称等信息。所以R语言里为我们提供了head()函数，在dplyr里面有一个实现类似功能的tbl_df()函数，分别使用效果如下。可以看到这个数据集包括了年、月、日、离开时间、计划到达时间、实际到达时间、延误时间等行。

# 第二部分 数据预览

```
> head(flights)
Source: local data frame [6 x 19]

   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
  (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1  2013     1     1      517            515         2      830            819        11
2  2013     1     1      533            529         4      850            830        20
3  2013     1     1      542            540         2      923            850        33
4  2013     1     1      544            545        -1     1004           1022       -18
5  2013     1     1      554            600        -6      812            837       -25
6  2013     1     1      554            558        -4      740            728        12
Variables not shown: carrier (chr), flight (int), tailnum (chr), origin (chr), dest
  (chr), air_time (dbl), distance (dbl), hour (dbl), minute (dbl), time_hour (time)
```

# 第二部分 数据预览

```
> tbl_df(flights)
Source: local data frame [336,776 x 19]

    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1   2013     1     1      517            515         2      830            819        11
2   2013     1     1      533            529         4      850            830        20
3   2013     1     1      542            540         2      923            850        33
4   2013     1     1      544            545        -1     1004           1022       -18
5   2013     1     1      554            600        -6      812            837       -25
6   2013     1     1      554            558        -4      740            728        12
7   2013     1     1      555            600        -5      913            854        19
8   2013     1     1      557            600        -3      709            723       -14
9   2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
..   ...   ...   ...      ...            ...       ...      ...            ...       ...
Variables not shown: carrier (chr), flight (int), tailnum (chr), origin (chr), dest
  (chr), air_time (dbl), distance (dbl), hour (dbl), minute (dbl), time_hour (time)
```

# 第二部分 数据预览

- head：只统计前六条，也只显示前六条数据
- table_df：信息更加详细统计全部数据，显示前十条数据

# 第三部分 数据筛选

以往我们对data.frame进行数据筛选是通过筛选索引，比如，我们想找出一月一日的数据就可以用一下：

- flights[flights$month == 1 & flights$day == 1, ]

在dplyr里面，则提供了一个filter()函数，可以更加简便的实现上述功能。

# 第三部分 数据筛选

```
filter(flights, month == 1, day == 1)

Source: local data frame [842 x 19]

    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1   2013     1     1      517            515         2      830            819        11
2   2013     1     1      533            529         4      850            830        20
3   2013     1     1      542            540         2      923            850        33
4   2013     1     1      544            545        -1     1004           1022       -18
5   2013     1     1      554            600        -6      812            837       -25
6   2013     1     1      554            558        -4      740            728        12
7   2013     1     1      555            600        -5      913            854        19
8   2013     1     1      557            600        -3      709            723       -14
9   2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
..   ...   ...   ...      ...            ...       ...      ...            ...       ...
Variables not shown: carrier (chr), flight (int), tailnum (chr), origin (chr), dest
  (chr), air_time (dbl), distance (dbl), hour (dbl), minute (dbl), time_hour (time)
```

# 第三部分 数据筛选

类似的，下面语句也经常用到

```
>filter(flights, month == 1 | month == 2)
```

```
>filter(flights, month > 6)
```

# 第四部分 数据重排序

除了filter()函数，dplyr还提供了一个arrange()函数可以帮助用户对于数据各行进行重新排序。

例如将所有数据按年月日的优先度进行row的重新排序

# 第四部分 数据重排序

```
arrange(flights, year, month, day)

Source: local data frame [336,776 x 19]

   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
  (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1  2013     1     1      517            515         2      830            819        11
2  2013     1     1      533            529         4      850            830        20
3  2013     1     1      542            540         2      923            850        33
4  2013     1     1      544            545        -1     1004           1022       -18
5  2013     1     1      554            600        -6      812            837       -25
6  2013     1     1      554            558        -4      740            728        12
7  2013     1     1      555            600        -5      913            854        19
8  2013     1     1      557            600        -3      709            723       -14
```

# 第四部分 数据重排序

也可以用desc关键字将航班延误时间做降序

```
arrange(flights, desc(arr_delay))

Source: local data frame [336,776 x 19]

    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1   2013     1     9      641            900      1301     1242           1530      1272
2   2013     6    15     1432           1935      1137     1607           2120      1127
3   2013     1    10     1121           1635      1126     1239           1810      1109
4   2013     9    20     1139           1845      1014     1457           2210      1007
5   2013     7    22      845           1600      1005     1044           1815       989
6   2013     4    10     1100           1900       960     1342           2211       931
7   2013     3    17     2321            810       911      135           1020       915
8   2013     7    22     2257            759       898      121           1026       895
9   2013    12     5      756           1700       896     1058           2020       878
10  2013     5     3     1133           2055       878     1250           2215       875
..   ...   ...   ...      ...            ...       ...      ...            ...       ...
```

# 第五部分 数据选择

选取数据框里面的部分列，这一功能以前也是利用索引实现的，dplyr则使用select函数可以更加便捷的实现选择功能。

例如选择这个数据集前三列(年月日)的数据。

# 第五部分 数据选择

```
select(flights, year, month, day)
select(flights, year:day)

Source: local data frame [336,776 x 3]

    year month   day
   (int) (int) (int)
1   2013     1     1
2   2013     1     1
3   2013     1     1
4   2013     1     1
5   2013     1     1
6   2013     1     1
7   2013     1     1
8   2013     1     1
9   2013     1     1
10  2013     1     1
..   ...   ...   ...
```

# 第五部分 数据选择

也可以利用distinct()函数根据某列的数值对重复行进行筛选。

例如择origin与dest组合互不相同的所有行的数据。

# 第五部分 数据选择

```
distinct(select(flights, origin, dest))

Source: local data frame [224 x 2]


   origin  dest
    (chr) (chr)
1     EWR   IAH
2     LGA   IAH
3     JFK   MIA
4     JFK   BQN
5     LGA   ATL
6     EWR   ORD
```

# 第六部分 数据变形

在dplyr包里，我们可以用mutate()函数直接利用已有的数据生成新的变量，这在使用相关分类和聚类算法的时候尤其好用。

还可以用transform()函数直接修改已经存在的列（变量）。如果只想保留新生成的列（变量），可以使用transmute()函数：

# 第六部分 数据变形

```
mutate(flights,gain = arr_delay - dep_delay, speed = distance/air_time * 60)

Source: local data frame [336,776 x 21]

    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   (int) (int) (int)    (int)          (int)     (dbl)    (int)          (int)     (dbl)
1   2013     1     1      517            515         2      830            819        11
2   2013     1     1      533            529         4      850            830        20
3   2013     1     1      542            540         2      923            850        33
4   2013     1     1      544            545        -1     1004           1022       -18
5   2013     1     1      554            600        -6      812            837       -25
6   2013     1     1      554            558        -4      740            728        12
7   2013     1     1      555            600        -5      913            854        19
8   2013     1     1      557            600        -3      709            723       -14
9   2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
..   ...   ...   ...      ...            ...       ...      ...            ...       ...
Variables not shown: carrier (chr), flight (int), tailnum (chr), origin (chr), dest
  (chr), air_time (dbl), distance (dbl), hour (dbl), minute (dbl), time_hour (time), gain
  (dbl), speed (dbl)
```

# 第六部分 数据选择

当然我们还可以用 transform()函数直接修改以生成的行列（变量）。

```
transform(flights,

  gain = arr_delay - delay,

  gain_per_hour = gain / (air_time / 60)

)
```

# 第六部分 数据选择

如果你只想保留新生成的行列（变量），可以使用 transmute()函数:

```
transmute(flights,
          gain = arr_delay - dep_delay,
          gain_per_hour = gain / (air_time / 60)
)
```

```
Source: local data frame [336,776 x 2]

    gain gain_per_hour
   (dbl)         (dbl)
1     9      2.378855
2    16      4.229075
```

# 第七部分 数据分组

在dplyr中我们使用group_by来分类数据，比如以下代码第一句便表示通过tailnum这个属性对航班数据进行分类。

```
by_tailnum <- group_by(flights, tailnum)
delay <- summarise(by_tailnum,
  count = n(),
  dist = mean(distance, na.rm = TRUE),
  delay = mean(arr_delay, na.rm = TRUE))
delay <- filter(delay, count > 20, dist < 2000)
```

# 第七部分 数据分组

```
by_tailnum <- group_by(flights, tailnum)

delay <- summarise(by_tailnum,

  count = n(),

  dist = mean(distance, na.rm = TRUE),

  delay = mean(arr_delay, na.rm = TRUE))

delay <- filter(delay, count > 20, dist < 2000)
```

# 第八部分 数据汇总

在dplyr包里我们使用summerise()函数进行数据汇总。下面的代码表示对平均离开时间的延误时间取平均，其中na.rm则表示去除所有含有缺失数据的行。
除此之外，我们还可以用sample_n() and sample_frac()函数随机选择计算汇总数据。

```
summarise(flights,
  delay = mean(dep_delay, na.rm = TRUE))

Source: local data frame [1 x 1]

     delay
     (dbl)
1 12.63907
```

# 第九部分 其他技巧

dplyr包里的一些小技巧

```
n(x) #x中行的数量
n_distinct(x): #x中不重复行的数量
first(x), last(x) #x中第一行与最后一行
```

# dplyr包

*谢谢*