

Data Transformation I

Introduction



- Read Chapter 3
- Goal: Their Data

 Your Data
- Covers:
 - Data Subsetting
 - Data Ordering
 - Variable Selecting
 - Variable Creating
- Help: dplyr Package in R

NYC Flights Meta Data



- Requirements:
 - > install.packages(nycflights13)
 - > library(nycflights13)
- All 2013 Flights from NYC
 - US Bureau of Trans. Statistics
 - To View all Data, Use > View(flights)

For more information, > ?flights

NYC Flights Meta Data



Preview Data: > flights

```
> flights
# A tibble: 336,776 x 19
                 day dep_time sched_dep_time dep_delay arr_time sched_arr_time
    year month
                          <int>
    2013
                           517
                                           515
                                                                830
                                                                                819
    2013
                           533
                                           529
                                                                850
                                                                                830
    2013
                                           540
                                                               923
                                                                                850
    2013
                                           545
                                                              1004
                                                                              1022
    2013
                                           600
                                                               812
                                                                                837
    2013
                                           558
                                                               740
                                                                                728
    2013
                                                               913
                                                                                854
                                           600
    2013
                           557
                                           600
                                                               709
                                                                                723
    2013
                                           600
                                                                838
                                                                                846
                           558
                                                                                745
    2013
                                           600
                                                               753
  ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
    carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
    air_time <db7>, distance <db7>, hour <db7>, minute <db7>, time_hour <dttm>
```

- Four Different Types of Variables
 - int = integer
 - dbl = double
 - chr = character
 - dttm = date and times
- Other Types of Variables
 - Igl = logical (TRUE or FALSE)
 - fctr = factor
 - date = dates

Basics of dplyr



- 5 Key Functions
 - filter() = Chooses Observations
 Based on Values
 - arrange() = Sorts Observations
 - select() = Chooses Variables
 - mutate() = Creates New Variables
 - summarize() = Generates
 Statistics From Data

Basics of dplyr



- Function Usage
 - First, Specify the Dataset
 - Next, Specify What to Do with the Data
 - Result is a New Dataset
- Powerful When Used With group_by() Function

Comparisons



Important Operators

- Less Than (<)
- Greater Than (>)
- Not Equal (!=)
- Equal (==)

Returns TRUE or FALSE

Comparisons



Numerical Precision

Problem

```
> x=1/49
> y=49
> x*y==1
[1] FALSE
> near(x*y,1)
[1] TRUE
```

Solution

```
> x*y
[1] 1
> near(x*y,1)
[1] TRUE
```

Logical Operators



Boolean Logic

- And (&)
- Or (|)
- Not (!)

Example

```
> #Basic
> x&y
[1] FALSE
> x | y
[1] TRUE
> ! x
[1] FALSE
> #Combined
> |x||y
[1] TRUE
> !(x&y)
[1] TRUE
> !x\&!y
[1] FALSE
```

Missing Values



- Represented by NA
 - Enduring Questions
 - To Impute or Not Impute
 - To Ignore or Not Ignore
 - Handling Should Be Explained
 - Be Careful When Performing Operations on Missing Data

Missing Values



```
> male.age=c(NA,20,21,35,22,NA)
> female.age=c(21,NA,23,33,22,NA)
  age.data=tibble(ma=male.age,fa=female.age)
  age.data
 A tibble: 6 x 2
           fa
     ma
  <db1> <db1>
     NA
           21
     20
           NA
          23
     21
     35
       33
     22
           22
     NA
           NA
> is.na(male.age)
    TRUE FALSE FALSE FALSE
> na.omit(age.data)
# A tibble: 3 x 2
           fa
     ma
  <db1> <db1>
     21
          23
     35 33
     22
          22
> mean(male.age)
[1] NA
> mean(male.age,na.rm=T)
[1] 24.5
```

filter()



- Used to Subset Observations Based on Their Values
 - Selects Row if TRUE
 - Removes Row if FALSE
- Examples:
 - All Flights from 9/13/2013 Out of LaGuardia Airport
- > filter(flights,month==9,day==13,origin =="LGA")
 - All Dec. or Nov. Flights
 - > filter(flights,month==11|month==12)
 - > filter(flights,month %in% c(11,12))

filter()



- Examples:
 - Don't Want Flights with Unusual Delays (> 120 min.)
- > filter(flights, !(arr_delay>120 | dep_delay>120))
- > filter(flights, arr_delay <= 120, dep_delay <= 120)</pre>
 - Want Flights with No Delays
- > filter(flights, dep_delay==0, arr_delay==0)
- > filter(flights, dep_delay==0 & arr_delay==0)

filter()



- Examples:
 - Want Flights Missing Air Time
 - > filter(flights, is.na(air_time))

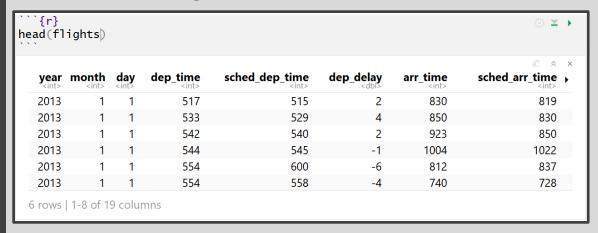
- Do not Want Flights Missing Air Time
 - > filter(flights, !is.na(air_time))

- Remove All Cases with Missing Values
 - > na.omit(flights)

arrange()



- Used to Sort Observations
- Raw Flight Data

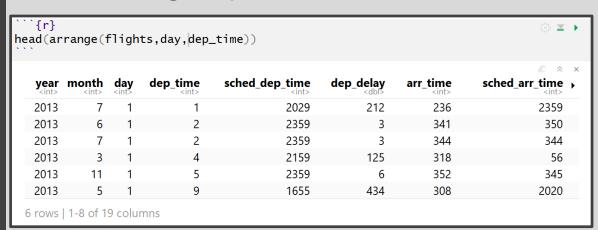


Sorted by Day, Month, Year (Time)

arrange()



Sorting Experiment

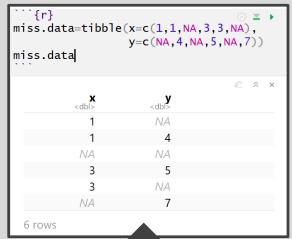


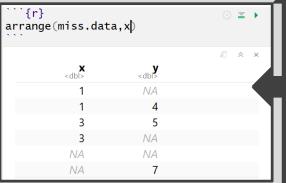
```{r} head(ar	range(1	fligh	ts,desc(day	/) ,dep_time))			⊕ ▼ ▶
year <int></int>	month <int></int>	day <int></int>	dep_time <int></int>	sched_dep_time	dep_delay	arr_time	sched_arr_time >
2013	1	31	1	2100	181	124	2225
2013	1	31	4	2359	5	455	444
2013	1	31	7	2359	8	453	437
2013	7	31	10	2359	11	344	340
2013	1	31	12	2250	82	132	7
2013	12	31	13	2359	14	439	437
	12   1-8 of 1			2359	14	439	437

arrange()



# Handling NA





		<i>□</i>
<b>X</b> <dbl></dbl>	<dbl></dbl>	
3	5	
3	NA	
1	NA	
1	4	
NA	NA	
NA	7	

```
> order(miss.data$x)
[1] 1 2 4 5 3 6
> order(desc(miss.data$x))
[1] 4 5 1 2 3 6
> is.na(miss.data$x)
[1] FALSE FALSE TRUE FALSE FALSE
[6] TRUE
> order(is.na(miss.data$x))
[1] 1 2 4 5 3 6
> order(is.na(miss.data$x),
+ decreasing=T)
[1] 3 6 1 2 4 5
```



- Used to Select Variables
- Why? Not All Variables are Created Equal
- Need to Know Variable Names

```
> names(flights)
 "month"
 "day"
 "year"
 "dep_time"
 "sched_dep_time" "dep_delay"
 "sched_arr_time" "arr_delay"
 "arr_time"
 "tailnum"
[10]
 "carrier"
 "flight"
 "origin"
 "dest"
 "air_time"
[13]
 "distance"
 "hour"
 "minute"
[16]
[19] "time_hour"
```



- Basic Examples
  - Select Only Year, Month, Day

```
> data1=select(flights,year,month,day)
> names(data1)
[1] "year" "month" "day"
```

 Select All Variables Between dep_time to arr_delay

Deselect Year, Month, and Day



#### Select Based on Column Index

```
> length(names(flights))
[1] 19
> data4=select(flights,c(1,3,8,12))
> names(data4)
[1] "year"
[2] "day"
[3] "sched_arr_time"
[4] "tailnum"
```

#### Deselect Based on Column Index

```
length(names(flights))
Γ1 19
> data5=select(flights,-c(1,3,8,12))
> names(data5)
 [1] "month"
 "dep_time"
 [3] "sched_dep_time"
 "dep_delay"
 "arr_time"
 [6]
 "arr_delay"
 "carrier"
 "flight"
 [8]
 [9]
 "origin"
[10]
 "dest"
 "air_time"
[11]
 "distance"
Γ12]
 "hour"
[13]
 "minute"
[14]
[15]
 "time_hour"
```



- Select Based on Text
  - starts_with("TEXT")

```
> data6=select(flights,starts_with("dep"))
> names(data6)
[1] "dep_time" "dep_delay"
```

ends_with("TEXT")

```
> data7=select(flights,ends_with("delay"))
> names(data7)
[1] "dep_delay" "arr_delay"
```

contains("TEXT")

```
> data8=select(flights,contains("ar"))
> names(data8)
[1] "year" "arr_time"
[3] "sched_arr_time" "arr_delay"
[5] "carrier"
```

Etc. AKA Others Exist



# Renaming Variables

# Can Use select()

```
> data9=select(flights,yr=year)
> names(data9)
[1] "yr"
```

# But Use rename()

```
> data10=rename(flights,yr=year)
> names(data10)
 [1] "yr"
 "month"
 "day"
 "dep_time"
 "sched_dep_time" "dep_delay"
 "arr_time"
 "sched_arr_time"
 "arr_delay"
 "carrier"
 "flight"
 "tailnum"
[11]
 "origin"
 "dest"
[13]
 "air_time"
 "distance"
 "hour"
 "minute"
[19]
 "time_hour"
```



# Reordering Variables

```
> head(flights)
A tibble: 6 x 19
 year month
 day dep_time sched_dep_time
 <int> <int> <int>
 <int>
 <int>
 2013
 1
 517
 515
 2013
 533
 529
 2013
 542
 540
 2013
 1
 545
 544
 2013
 554
 600
 2013
 554
 558
 ... with 14 more variables: dep_delay <db1>,
 arr_time <int>, sched_arr_time <int>,
 arr_delay <dbl>, carrier <chr>,
 flight <int>, tailnum <chr>, origin <chr>,
 dest <chr>, air_time <db1>, distance <db1>,
 hour <db1>, minute <db1>, time_hour <dttm>
 data11=select(flights,dep_time,arr_time,
 air_time.everything())
> head(data11)
A tibble: 6 x 19
 dep_time arr_time air_time year month
 <db1> <int> <int> <int>
 <int>
 <int>
 517
 830
 227 2013
 1
 533
 850
 227 2013
 542
 923
 160 2013
 544
 1004
 183 2013
 554
 812
 116 2013
 1
 554
 740
 150 2013
 ... with 13 more variables:
 sched_dep_time <int>, dep_delay <dbl>,
 sched_arr_time <int>, arr_delay <dbl>,
 carrier <chr>, flight <int>, tailnum <chr>,
 origin <chr>, dest <chr>, distance <db1>,
 hour <db1>, minute <db1>, time_hour <dttm>
```

Closing



# Disperse and Make Reasonable Decisions