Data Manipulation: Summarize( ) & Group by( )

Pilsung Kang

School of Industrial Management Engineering

Korea University

dplyr: Summarize( )

- Summarize( )
  - ✓ Applies a function to variable
  - ✓ Arguments
    - Data frame
    - Definition of a summary statistic
  - ✓ Example: summarize(data*, mean_weight = mean(weight))
    - data* must be a tibble
  - ✓ Creating summary statistics from a complex data set is obviously a crucial task in data analysis
  - ✓ In dplyr this is done with the function summarise() that creates a new data frame with a single row with statistics
  - ✓ The syntax is the same as mutate:

```
summarise(df, AverageBmi = mean(bmi))
```

dplyr: Summarize( )

- Summarize( )
  - ✓ Determine the shortest and longest distance flown and save statistics to min_dist and max_dist
  - ✓ Determine the longest distance for diverted flights, save statistic to max_div

```
# Determine the shortest and longest distance flown and save statistics to
min_dist and max_dist
summarize1 <- summarize(hflights, min_dist = min(Distance),
                        max_dist = max(Distance))
summarize1
# Determine the longest distance for diverted flights, save statistic to
max_div
summarize2 <- summarize(filter(hflights, Diverted==1),
                        max_div = max(Distance))
summarize2
```

```
> summarize1
  min_dist max_dist
1       79     3904
```

```
> summarize2
  max_div
1    3904
```

dplyr: Summarize( )

- Summarize( )
  - ✓ Aggregate functions
    - We can use any function so long as the function can take a vector of data and return a single number

      - `min(x)` - minimum value of vector `x`.
      - `max(x)` - maximum value of vector `x`.
      - `mean(x)` - mean value of vector `x`.
      - `median(x)` - median value of vector `x`.
      - `quantile(x, p)` - pth quantile of vector `x`.
      - `sd(x)` - standard deviation of vector `x`.
      - `var(x)` - variance of vector `x`.
      - `IQR(x)` - Inter Quartile Range (IQR) of vector `x`.
      - `diff(range(x))` - total range of vector `x`.

dplyr: Summarize( )

- Summarize( )
  - ✓ Aggregate functions

```
# Aggregate functions in basic R
agg1 <- filter(hflights2, !is.na(ArrDelay))
agg2 <- summarize(agg1,
                  earliest = min(ArrDelay),
                  average = mean(ArrDelay),
                  latest = max(ArrDelay),
                  sd = sd(ArrDelay))
agg2
```

```
> agg2
# A tibble: 1 x 4
  earliest average latest    sd
     <int>   <dbl>  <int> <dbl>
1      -70    7.09    978  30.7
```

dplyr: Summarize( )

- Summarize( )
  - ✓ Aggregate functions
    - dplyr provides several helpful aggregate functions of its own, in addition to the ones that are already defined in R

- `first(x)` - The first element of vector `x`.
- `last(x)` - The last element of vector `x`.
- `nth(x, n)` - The nth element of vector `x`.
- `n()` - The number of rows in the data.frame or group of observations that summarise() describes.
- `n_distinct(x)` - The number of unique values in vector `x`.

dplyr: Summarize( )

- Summarize( )
  - ✓ Aggregate functions: dplyr provides several helpful aggregate functions of its own, in addition to the ones that are already defined in R

```
# Additional aggregate functions provided by dplyr
agg3 <- summarise(hflights2, n_obs = n(),
                  n_carrier = n_distinct(UniqueCarrier),
                  n_dest = n_distinct(Dest),
                  dest100 = nth(Dest, 100))
agg3
```

```
> agg3
# A tibble: 1 x 4
   n_obs n_carrier n_dest dest100
   <int>     <int>  <int> <chr>
1 227496        15    116 DFW
```

dplyr: Summarize( )

- Summarize( )
  - ✓ Aggregate functions: dplyr provides several helpful aggregate functions of its own, in addition to the ones that are already defined in R

```
# Calculate the summarizing statistics for flights flown by American
Airlines (carrier code AA)
AA <- filter(hflights2, UniqueCarrier == "AA")
agg4 <- summarise(AA, n_flights = n(),
                      n_canc = sum(Cancelled == 1),
                      p_canc = mean(Cancelled == 1) * 100,
                      avg_delay = mean(ArrDelay, na.rm = TRUE))
agg4
```

```
> agg4
# A tibble: 1 x 4
  n_flights n_canc p_canc avg_delay
      <int>  <int>  <dbl>     <dbl>
1      3244     60   1.85     0.892
```

dplyr: Group by( )

- Group by( )
  - ✓ Groups data in the table by an attribute
  - ✓ Arguments
    - Data frame
    - Factor variable to group by
  - ✓ Example: group_by(surveys, sex)
  - ✓ Very often we are interested in computing summary statistics for each value of a given variable
  - ✓ For instance, we might want to compute the average bmi separately for men and women or for each age category
  - ✓ In this case we can use group_by() to create a *grouped* data frame in which any following operations will be done accordingly *by group*:

```
group_by(df, sex) %>% summarise(mean(bmi))
```

dplyr: Group by( )

- Group by( )
  - ✓ The average departure and arrival delays for each day of the week

```
# The average departure and arrival delays for each day of the week
hflights2 %>% group_by(DayOfWeek) %>%
    summarise(AverageArrDelay = mean(ArrDelay, na.rm = TRUE),
              AverageDepDelay = mean(DepDelay, na.rm = TRUE))
```

```
> hflights2 %>% group_by(DayOfWeek) %>%
+   summarise(AverageArrDelay = mean(ArrDelay, na.rm = TRUE),
+             AverageDepDelay = mean(DepDelay, na.rm = TRUE))
# A tibble: 7 x 3
  DayOfWeek AverageArrDelay AverageDepDelay
      <int>           <dbl>           <dbl>
1         1            8.26            10.0
2         2            5.55             7.59
3         3            5.53             8.08
4         4            9.80            12.4
5         5            7.29             9.88
6         6            5.75             7.77
7         7            6.95             9.78
```

dplyr: Group by( )

- Group by( )
  - ✓ The average departure and arrival delays for each day of the week
  - ✓ With basic R syntax without dplyr

```
# Note how more immediate is feels compared to basic R
AverageArrDelay <- tapply(hflights$ArrDelay, hflights$DayOfWeek,
                          mean, na.rm = TRUE)
AverageDepDelay <- tapply(hflights$DepDelay, hflights$DayOfWeek,
                          mean, na.rm = TRUE)
cbind(sort(unique(hflights$DayOfWeek)), AverageArrDelay, AverageDepDelay)
```

```
> cbind(sort(unique(hflights$DayOfWeek)), AverageArrDelay, AverageDepDelay)
     AverageArrDelay AverageDepDelay
1 1        8.255831       10.025682
2 2        5.551781        7.591971
3 3        5.533013        8.083891
4 4        9.797332       12.404041
5 5        7.291188        9.877408
6 6        5.746582        7.772742
7 7        6.950572        9.777305
```

dplyr: Group by( )

- Group by( )
  - ✓ We rank airline companies according to their average departure delay

```
hflights2 %>% filter(!is.na(DepDelay), DepDelay > 0) %>%
    # we keep only flights with a departure delay
    group_by(UniqueCarrier) %>%
    summarise(avg = mean(DepDelay)) %>%
    # average departure delay for each company
    mutate(rank = rank(avg)) %>%
    arrange(rank)
```

```
# A tibble: 15 x 3
   UniqueCarrier   avg  rank
   <chr>         <dbl> <dbl>
 1 CO             17.9     1
 2 AS             20.8     2
 3 WN             21.9     3
 4 F9             22.7     4
 5 YV             24.5     5
 6 OO             24.6     6
 7 AA             24.7     7
 8 US             26.5     8
 9 XE             26.9     9
10 UA             28.8    10
11 DL             32.4    11
12 FL             33.4    12
13 MQ             37.9    13
14 B6             43.5    14
15 EV             49.3    15
```

dplyr: Group by( )

- Group by( )
  - ✓ Note how complicate it would have been not to use the %>% operator in the previous example:

```
hflights2 <- group_by(filter(hflights2, !is.na(DepDelay), DepDelay > 0),
             UniqueCarrier)

arrange( mutate(summarise(hflights2, avg = mean(DepDelay)),
rank = rank(avg)
  ),
  rank
)
```

dplyr: Group by( )

- Group by( )
  - ✓ Arrange the UniqueCarrier with the delay proportion and their rank

```
# Arrange the UniqueCarrier with the delay proportion and their rank
hflights2 %>%
    group_by(UniqueCarrier) %>%
    filter(!is.na(ArrDelay)) %>%
    summarise(p_delay = mean(ArrDelay > 0)) %>%
    mutate(rank = rank(p_delay)) %>%
    arrange(rank)
```

```
# A tibble: 15 x 3
   UniqueCarrier p_delay  rank
   <chr>           <dbl> <dbl>
 1 AA              0.303     1
 2 FL              0.311     2
 3 US              0.327     3
 4 EV              0.368     4
 5 MQ              0.370     5
 6 DL              0.387     6
 7 B6              0.395     7
 8 AS              0.437     8
 9 WN              0.464     9
10 YV              0.474    10
11 CO              0.491    11
12 XE              0.494    12
13 UA              0.496    13
14 OO              0.535    14
15 F9              0.556    15
```

dplyr: Group by( )

- Group by( )
  - ✓ Arrange the UniqueCarrier with the average arrival delay time with their rank

```
# Arrange the UniqueCarrier with the average arrival delay time with their
rank
hflights2 %>%
    group_by(UniqueCarrier) %>%
    filter(!is.na(ArrDelay), ArrDelay > 0) %>%
    summarise(avg = mean(ArrDelay)) %>%
    mutate(rank = rank(avg)) %>%
    arrange(rank)
```

```
# A tibble: 15 x 3
   UniqueCarrier   avg  rank
   <chr>         <dbl> <dbl>
 1 YV             18.7     1
 2 F9             18.7     2
 3 US             20.7     3
 4 CO             22.1     4
 5 AS             22.9     5
 6 OO             24.1     6
 7 XE             24.2     7
 8 WN             25.3     8
 9 FL             27.9     9
10 AA             28.5    10
11 DL             32.1    11
12 UA             32.5    12
13 MQ             38.8    13
14 EV             40.2    14
15 B6             45.5    15
```

dplyr: Group by( )

- Group by( )
  - ✓ Which plane (by tail number) flew out of Houston the most times? How many times?

```
# Which plane (by tail number) flew out of Houston the most times? How many
times?
hflights2 %>%
    group_by(TailNum) %>%
    summarise(n = n()) %>%
    filter(n == max(n))
```

```
# A tibble: 1 x 2
  TailNum     n
  <chr>   <int>
1 N14945    971
```

dplyr: Group by( )

- Group by( )
  - ✓ How many airplanes only flew to one destination from Houston?

```
# How many airplanes only flew to one destination from Houston?
hflights2 %>%
    group_by(TailNum) %>%
    summarise(ndest = n_distinct(Dest)) %>%
    filter(ndest == 1) %>%
    summarise(nplanes = n())
```

```
# A tibble: 1 x 1
  nplanes
    <int>
1    1526
```

dplyr: Group by( )

- Group by( )
  - ✓ Find the most visited destination for each carrier

```
# Find the most visited destination for each carrier
hflights2 %>%
    group_by(UniqueCarrier, Dest) %>%
    summarise(n = n()) %>%
    mutate(rank = rank(desc(n))) %>%
    filter(rank == 1)
```

```
# A tibble: 15 x 4
# Groups:   UniqueCarrier [15]
   UniqueCarrier Dest      n  rank
   <chr>         <chr> <int> <dbl>
 1 AA            DFW    2105     1
 2 AS            SEA     365     1
 3 B6            JFK     695     1
 4 CO            EWR    3924     1
 5 DL            ATL    2396     1
 6 EV            DTW     851     1
 7 F9            DEN     837     1
 8 FL            ATL    2029     1
 9 MQ            DFW    2424     1
10 OO            COS    1335     1
11 UA            SFO     643     1
12 US            CLT    2212     1
13 WN            DAL    8243     1
14 XE            CRP    3175     1
15 YV            CLT      71     1
```

dplyr: Group by( )

- Group by( )
  - ✓ Find the carrier that travels to each destination the most

```
# Find the carrier that travels to each destination the most
hflights2 %>%
    group_by(Dest, UniqueCarrier) %>%
    summarise(n = n()) %>%
    mutate(rank = rank(desc(n))) %>%
    filter(rank == 1)
```

```
# A tibble: 116 x 4
# Groups:   Dest [116]
   Dest  UniqueCarrier     n  rank
   <chr> <chr>         <int> <dbl>
 1 ABQ   WN             1019     1
 2 AEX   XE              724     1
 3 AGS   CO                1     1
 4 AMA   XE             1297     1
 5 ANC   CO              125     1
 6 ASE   OO              125     1
 7 ATL   DL             2396     1
 8 AUS   CO             2645     1
 9 AVL   XE              350     1
10 BFL   OO              504     1
# ... with 106 more rows
```


ANY
questions ?