

# Guided Quiz: Sorting Data Frames and Tables By Columns

Mert Nuhoglu  
July 22, 2015

- What is guided quiz?
  - Question
  - Answer
  - Explanation

# Guided Quiz: Sorting Data Frames and Tables By Columns

# How to sort a dataframe by columns

```
df = data.frame(  
  name = c("Ali", "Olli", "Billy"),  
  age = c(15, 30, 20),  
  state = c("NY", "CA", "NY")  
)
```

# Sort by one column

- Sort the dataframe by its name column
- Ascending order

```
df = data.frame(  
  name = c("Ali", "Olli", "Billy"),  
  age = c(15, 30, 20),  
  state = c("NY", "CA", "NY")  
)  
df
```

```
##      name age state  
## 1   Ali  15    NY  
## 2  Olli  30    CA  
## 3 Billy  20    NY
```

# Solution

```
df[ order(df$name), ]
```

```
##      name age state  
## 1   Ali  15    NY  
## 3 Billy  20    NY  
## 2  Olli  30    CA
```

# order() function

```
order( c("Apple", "IBM", "Microsoft") )
```

```
## [1] 1 2 3
```

```
order( c("IBM", "Microsoft", "Apple" ) )
```

```
## [1] 3 1 2
```

# Solution

```
df[ order(df$name), ]
```

```
##      name age state  
## 1   Ali  15    NY  
## 3 Billy  20    NY  
## 2  Olli  30    CA
```

# Question: Sort using datatable

- Use datatable instead of dataframe

```
dt = data.table(df)
dt
```

```
##      name age state
## 1:   Ali  15    NY
## 2:  Olli  30    CA
## 3: Billy  20    NY
```



# Solution

```
dt[ order(dt$name) ]
```

```
##      name age state
## 1:   Ali  15   NY
## 2: Billy  20   NY
## 3:  Olli  30   CA
```

instead of

```
df[ order(df$name), ]
```

## Question: Reverse the order

- Descending order instead of ascending order
- That is: Letter Z should come first

# Solution

```
df[ order(-df$name), ]

## Warning in Ops.factor(df$name): '-' not meaningful for factors

##      name age state
## 1   Ali  15    NY
## 2  Olli  30    CA
## 3 Billy  20    NY

dt[ order(-dt$name), ]

##      name age state
## 1:  Olli  30    CA
## 2: Billy  20    NY
## 3:   Ali  15    NY
```

## Solution 2

- Which one is first?

5 10 20

- Now?

-5 -10 -20

## Question: Sort using dplyr style

- Sort using dplyr style
- Use pipes `%>%` and `arrange` functions

# Solution

```
dt %>%  
  arrange( name )
```

```
##      name age state  
## 1:   Ali  15    NY  
## 2: Billy  20    NY  
## 3:  Olli  30    CA
```

```
df %>%  
  arrange( name )
```

```
##      name age state  
## 1   Ali  15    NY  
## 2 Billy  20    NY  
## 3  Olli  30    CA
```

## Solution 2

- With pipes

```
dt %>%  
  arrange( name )  
  
##      name age state  
## 1:   Ali  15    NY  
## 2: Billy  20    NY  
## 3: Olli  30    CA
```

- Without pipes

```
arrange( dt, name )  
  
##      name age state  
## 1:   Ali  15    NY  
## 2: Billy  20    NY  
## 3: Olli  30    CA
```

# Solution

```
dt %>%  
  arrange( name )
```

```
##      name age state  
## 1:   Ali  15    NY  
## 2: Billy  20    NY  
## 3:  Olli  30    CA
```

```
df %>%  
  arrange( name )
```

```
##      name age state  
## 1   Ali  15    NY  
## 2 Billy  20    NY  
## 3  Olli  30    CA
```



# Question: Using with and base order functions

- Use with() function
- Use base::order function
- Encapsulate the columns to order inside with

# Solution

```
df[ with(df, order(name)), ]
```

```
##      name age state
## 1   Ali  15    NY
## 3 Billy  20    NY
## 2  Olli  30    CA
```

Compare the ordinary way:

```
df[ order(-df$name), ]
```

```
## Warning in Ops.factor(df$name): '-' not meaningful for factors
```

```
##      name age state
## 1   Ali  15    NY
## 2  Olli  30    CA
## 3 Billy  20    NY
```

# Reverse the order in dplyr style

Sort using dplyr style. Reverse the order.

# Solution

```
dt %>%  
  arrange( desc(name) )  
  
##      name age state  
## 1:  Olli  30    CA  
## 2: Billy  20    NY  
## 3:   Ali  15    NY
```

# Sort by two columns

- Sort by two columns: state (ascending) and age (descending)
- Use dataframe and base order()

# Solution

```
df[ order(df$state, -df$age), ]
```

```
##      name age state  
## 2  Olli  30    CA  
## 3 Billy  20    NY  
## 1   Ali  15    NY
```

# Use datatable

- Same question
- Use datatable instead of dataframe
- Sort by two columns: state (ascending) and age (descending)

# Solution

- No comma sign

```
dt[ order(dt$state, -df$age) ]
```

```
##      name age state
## 1:  Olli  30    CA
## 2: Billy  20    NY
## 3:   Ali  15    NY
```



# Use dplyr functions

- Same question
- Use dplyr style
- Sort by two columns: state (ascending) and age (descending)

# Solution

```
dt %>%  
  arrange( state, desc(age) )  
  
##      name age state  
## 1:  Olli  30    CA  
## 2: Billy  20    NY  
## 3:   Ali  15    NY
```

# Conclusion

- Which style is better?
- Personal tastes
- My choice: dplyr
- Declarative programming style:
  - What to do?
  - Not how to do it?